**LỜI NÓI ĐẦU**

Là sinh viên của khoa Điện tử Viễn thông của trường Học viện Công nghệ Bưu chính Viễn thông, em cảm thấy rất vinh dự và tự hào khi đã được ngồi học, trải nghiệm tại một môi trường giáo dục bổ ích, tiến tiến và năng động.

Trong bốn tuần thực tập tại công ty cổ phần đầu tư Reway Group, em đã nhận được sự chỉ bảo, giúp đỡ tận tình của mentor và supervisor cùng các anh chị đồng nghiệp.

Trước hết em xin cảm ơn tất cả các anh chị đồng nghiệp đã giúp đỡ nhiệt tình trong quá trình thực tập tại công ty. Vì mới ra trường, chưa có nhiều kinh nghiệm nhưng mọi người đã nhiệt tình chỉ dạy, giúp đỡ em để em có thể lĩnh hội được những công nghệ mới nhất, những kiến thức thực tế, cách làm việc chuyên nghiệp.

Đặc biệt, em xin và tỏ lòng biết ơn sâu sắc tới anh Trần Văn Bách (mentor) người đã hướng dẫn em trong suốt quá trình thực tập.

Em xin gửi lời cảm ơn chân thành tới thầy cô, gia đình, bạn bè đặc biệt là các thành viên thuộc tập thể lớp D16VT5 đã động viên, đóng góp ý kiến và giúp đỡ em trong quá trình học tập, nghiên cứu và hoàn thành thực tập.

Em xin gửi lời cảm ơn tới tất cả các thầy cô giáo trong khoa viễn thông 1 đã chỉ dẫn em, cho em định hướng và cũng cấp cho em những kiến thức bổ ích để khi tiếp xúc với công nghệ mới em không bị bỡ ngỡ.

Cuối cùng, em xin kính chúc các anh chị đồng nghiệp, quý thầy, cô và gia

đình luôn dồi dào sức khoẻ, thành công trong sự cuộc sống. Chúc các bạn sinh viên luôn

luôn phấn đấu và thành công!

Em xin chân thành cảm ơn!

Hà Nội, Ngày 20 tháng 9 năm 2020

Sinh viên thực hiện

MỤC LỤC

[CHƯƠNG 1: KHÁI QUÁT VỀ MÃ QR 3](#_Toc58162353)

[1.1 Giới thiệu về mã QR 3](#_Toc58162354)

[1.2 Phân loại 4](#_Toc58162355)

[1.2.1 Mã QR tĩnh là gì? 4](#_Toc58162356)

[1.2.2 Mã QR biến đổi là gì? 4](#_Toc58162357)

[1.2.3 Ưu điểm và nhược điểm 5](#_Toc58162358)

[1.3 Cấu trúc của mã QR 5](#_Toc58162359)

[1.3.1 Cell (Ngôn ngữ) 6](#_Toc58162360)

[1.3.2 Hoa văn định vị 6](#_Toc58162361)

[1.3.3 Timming pattern (Khung thời gian) 6](#_Toc58162362)

[1.3.4 Alignment pattern (Ký hiệu căn chỉnh) 7](#_Toc58162363)

[1.3.5 Thông tin format (Chức năng sửa chữa lỗi) 7](#_Toc58162364)

[1.4 Khả năng lưu trữ và sửa lỗi 7](#_Toc58162365)

[1.4.1 Khả năng lưu trữ 7](#_Toc58162366)

[1.4.2 Khả năng sửa lỗi 8](#_Toc58162367)

[1.5 So sánh QRCode và BarCode 10](#_Toc58162368)

[1.5.1 BarCode là gì? 10](#_Toc58162369)

[1.5.2 Sự khác biệt giữa QRCode và BarCode 10](#_Toc58162370)

[1.6 Ưu điểm và nhược điểm của mã QR 11](#_Toc58162371)

[1.6.1 Ưu điểm 11](#_Toc58162372)

[1.6.2 Nhược điểm 11](#_Toc58162373)

[1.7 Ứng dụng 12](#_Toc58162374)

[CHƯƠNG 2: THIẾT KẾ VÀ BẢO MẬT MÃ QR 14](#_Toc58162375)

[2.1 Thiết kế mã QR 14](#_Toc58162376)

[2.2 Thuật toán mã hoá mã QR 14](#_Toc58162377)

[2.2.1 Tạo một chuỗi nhị phân 14](#_Toc58162378)

[2.2.2 Tạo bộ mã sửa lỗi 16](#_Toc58162379)

[2.2.3 Chọn mẫu mặt nạ tốt nhất 18](#_Toc58162380)

[2.3 Bảo mật mã QR với thông tin ba lớp sử dụng mã Hamming 26](#_Toc58162381)

[2.3.1 Mã Hamming 27](#_Toc58162382)

[2.3.2 Mô phỏng bảo mật 27](#_Toc58162383)

[CHƯƠNG 3: NỀN TẢNG REACT NATIVE VÀ LẬP TRÌNH THỬ NGHIỆM 30](#_Toc58162384)

[3.1 Tổng quan về lập trình ứng dụng di động React Native 30](#_Toc58162385)

[**3.1.1 Native và Hybird là gì?** 30](#_Toc58162386)

[**3.1.2. Tại sao phải là React Native?** 30](#_Toc58162387)

[3.1.3 Ưu điểm và nhược điểm 31](#_Toc58162388)

[**3.1.4 Công cụ hỗ trợ lập trình React Native** 31](#_Toc58162389)

[3.2 Thành phần chính của React Native 37](#_Toc58162390)

[3.2.1 Components 37](#_Toc58162391)

[3.2.2 Props 38](#_Toc58162392)

[3.2.3 State 40](#_Toc58162393)

[3.3 Kết nối API, Redux và Firebase vào dự án thực tế 41](#_Toc58162394)

[3.3.1 API là gì? 41](#_Toc58162395)

[3.3.2 Ưu và nhược điểm của API 42](#_Toc58162396)

[3.2.3 Cách dùng API trong dự án thực tế 42](#_Toc58162397)

[3.4 Kết nối Redux vào dự án Lotus – Quản lý kho 44](#_Toc58162398)

[3.4.1 Redux là gì và các thành phần của Redux 44](#_Toc58162399)

[3.4.2 Cài đặt, cấu hình và cách dùng Redux 44](#_Toc58162400)

[3.5 Kết nối Firebase vào dự án Lotus – Quản lý kho 47](#_Toc58162401)

[3.5.1 Firebase là gì? 47](#_Toc58162402)

[3.5.2 Các tính năng chính của Firebase 47](#_Toc58162403)

[3.5.3. Cài đặt, cấu hình và cách dùng Firebase 48](#_Toc58162404)

[3.6 Cách đẩy code lên hệ điều hành ios, android dự án Lotus – Quản lý kho 49](#_Toc58162405)

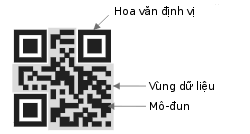
[KẾT LUẬN 51](#_Toc58162406)

**Nghiên cứu mã QR và xây dựng ứng dụng bằng nền tảng React Native**

# **CHƯƠNG 1: KHÁI QUÁT VỀ MÃ QR**

# **1.1 Giới thiệu về mã QR**

Mã QR bây giờ không hẳn là quá xa lạ, nó bắt đầu xuất hiện khắp nơi như nhãn bìa sản phẩm, và gần như là phuơng thức nhận diện chủ yếu cho ứng dụng di động. Vậy mã QR là gì? Mã QR hay mã hai chiều là một [mã vạch ma trận](https://vi.wikipedia.org/wiki/M%C3%A3_v%E1%BA%A1ch) (hay mã vạch hai chiều, mã phản hồi nhanh) được phát triển bởi công ty Denso Wave (Nhật Bản) vào năm 1994. Chữ "QR" xuất phát từ "Quick Response", trong tiếng Anh có nghĩa là *đáp ứng nhanh*, vì người tạo ra nó có ý định cho phép mã được giải mã ở tốc độ cao. Các mã QR được sử dụng phổ biến nhất ở [Nhật Bản](https://vi.wikipedia.org/wiki/Nh%E1%BA%ADt_B%E1%BA%A3n), [Trung Quốc](https://vi.wikipedia.org/wiki/Trung_Qu%E1%BB%91c), và hiện là loại mã hai chiều thông dụng nhất ở Nhật Bản.



*Hình 1: Cấu trúc của một ký hiệu mã QR*

Mặc dù lúc đầu mã QR được dùng để theo dõi các bộ phận trong sản xuất xe hơi, hiện nay nó được dùng trong quản lý kiểm kê ở nhiều ngành khác nhau. Gần đây hơn, phần mềm đọc mã QR đã được cài vào điện thoại di động có gắn camera (camera phone) ở Nhật. Điều này đưa đến các ứng dụng mới và đa dạng hướng về người tiêu dùng, nhằm làm nhẹ nhàng việc nhập dữ liệu vào điện thoại di động, vốn không hấp dẫn mấy. Mã QR cũng được thêm vào danh thiếp, làm đơn giản đi rất nhiều việc nhập dữ kiện cá nhân của người mới quen vào sổ địa chỉ trên điện thoại di động.

Mã QR đã và đang ngày càng phổ biến trong kỉ nguyên phát triển mạnh mẽ của smartphone. Có rất nhiều ứng dụng trên CHPlay hay AppStore free dùng để đọc mã QR. Vì vậy, chỉ cần nhìn thấy mã QR, rút smartphone ra và scan, ta sẽ ngay lập tức biết được thông tin trên đó ghi gì. Và ngày nay, các app còn tích hợp sẵn các action ngay khi scan được thông tin, như chuyển hướng vào browser nếu thông tin là website URL, gọi điện thoại nếu là số phone, ...

Người dùng có chương trình thu tín hiệu (capture program) và máy tính có [giao diện RS-232C](https://vi.wikipedia.org/w/index.php?title=Giao_di%E1%BB%87n_RS-232C&action=edit&redlink=1) có thể dùng [máy quét ảnh](https://vi.wikipedia.org/wiki/M%C3%A1y_qu%C3%A9t_%E1%BA%A3nh) (scanner) để thu dữ liệu. Tiêu chuẩn Nhật Bản cho các mã QR, [JIS](https://vi.wikipedia.org/w/index.php?title=JIS&action=edit&redlink=1) X 0510, được công bố vào tháng 1 năm 1999, và Tiêu chuẩn Quốc tế ISO tương ứng, ISO/IEC18004, được chấp thuận vào tháng 6 năm 2000.

# **1.2 Phân loại**

Mã QR được chia làm hai loại là mã QR tĩnh và mã QR động, có thể xem sự so sánh mã QR động và mã QR tĩnhdưới đây để tìm hiểu mỗi loại với những ưu điểm và nhược điểm riêng.

## 1.2.1 Mã QR tĩnh là gì?

Mã QR tĩnh hoạt động theo nguyên tắc lưu trữ dữ liệu trực tiếp qua hình thức văn bản dẫn thẳng đến trang web mà không qua các liên kết thứ cấp. Mã QR tĩnh chỉ dùng để lưu trữ những thông tin một chiều, cố định và không thể thay đổi được và nếu muốn cập nhật thay đổi hoặc chỉnh sửa thông tin ta chỉ có cách là thay đổi mã khác. Điều này có lợi khi doanh nghiệp muốn lưu trữ thông tin mang tính vĩnh cửu và không cần thay đổi.

## 1.2.2 Mã QR biến đổi là gì?

Mã QR biến đổi là loại mã có thể chỉnh sửa được ngay cả sau khi chúng đã được in, do vậy các doanh nghiệp có thể thay đổi hoặc cập nhật tin tức mới nhất về sản phẩm cũng như thông tin về doanh nghiệp của mình theo từng thời điểm khác nhau. Mã QR biến đổi sử dụng đường link URL ngắn để chuyển hướng người dùng đến trang đích mong muốn, cho phép doanh nghiệp có thể thu thập số liệu thống kê về các số quét, địa điểm, thời gian và hệ điều hành được sử dụng.

## 1.2.3 Ưu điểm và nhược điểm

**1.2.3.1 Ưu điểm**

Mã QR biến đổi:Có thể kết nối với hệ thống thông tin điện tử**,**chỉnh sửa các thông tin trong mã**,**theo dõi, thống kê quá trình quét mã, định danh thông tin chi tiết người quét mã, thu thập, phân tích dữ liệu người quét mã. Điều này đem đến cho doanh nghiệp và khách hàng sự an tâm trong việc xác thực hàng hóa, đưa ra thông tin chi tiết về sản phẩm có phải hàng chính hãng hay không. Có thể trả về thông tin của người xác thực lần đầu, cũng như thay đổi trạng thái của sản phẩm trong những lần xác thực sau. Ngăn chặn hiệu quả việc làm giả, làm nhái sản phẩm đem lại sự an toàn cho người dùng.

Mã QR tĩnh:Loại mã QR này được sử dụng phổ biến hơn, được nhiều người sử dụng hơn bởi ta có thể tạo chúng miễn phí và không cần bất kỳ yêu cầu công nghệ nào. Nếu có một liên kết web hoặc địa chỉ email được mã hóa thì thiết bị sẽ nhận được thông tin về những đường link được mã hóa này, tuy nhiên để đến được đường link đó hoặc gửi một email cần phải có kết nối Internet.

**1.2.3.2 Nhược điểm**

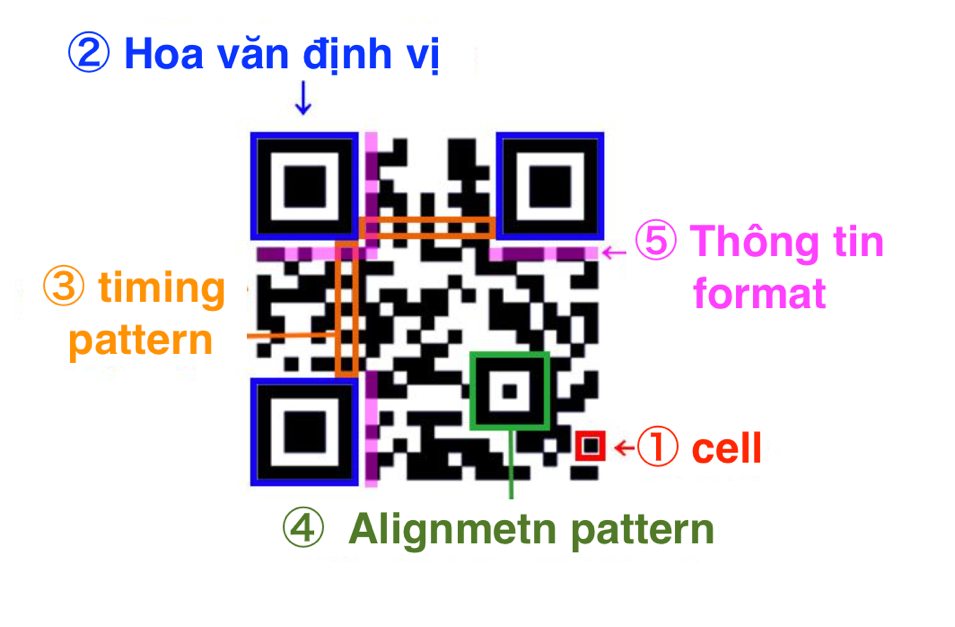
Mã QR biến đổi:doanh nghiệp muốn tạo ra phải mất phí cao và sử dụng công nghệ cao, đồng thời khi muốn thay đổi thông tin về sản phẩm cũng như thông tin về doanh nghiệp phải vào code. Khi vào trang đích mà ta mong muốn phải thông qua các liên kết thứ cấp mới có thể truy cập được, có nghĩa là nó có tính bảo mật hơn nhưng người tiêu dùng sẽ mất thời gian truy cập lâu hơn. Theoso sánh mã QR biến đổi và mã QR tĩnhthì nhược điểm của mã QR biến đổi là không đáng kể và nó sẽ được lựa chọn nhiều hơn.

Mã QR tĩnh:Trong xác thực hàng hoá, sản phẩm thì mã QR tĩnh chỉ có thể trả về thông tin sản phẩm mà mình xác thực, không thể đưa ra thông tin chi tiết về người xác thực cũng như ngày giờ xác thực. Quan trọng nhất là mã QR tĩnh không thể thay đổi trạng thái xác thực sản phẩm, điều này gây nguy hiểm cho người dùng khi những kẻ xấu muốn sao y bản chính mã QR sản phẩm chính hãng để làm nhái hoặc làm giả sản phẩm.

Như vậy người dùng sẽ không thể nhận diện được ai là người xác thực sản phẩm lần đầu, hoặc sản phẩm đã qua sử dụng hay chưa đồng nghĩa với việc truy xuất nguồn gốc sản phẩm này không thể chính xác được. Do không thể thay đổi thông tin nên mã QR tĩnh thể cập nhật các thông tin khuyến mãi, ưu đãi của doanh nghiệp vì vậy không thể đem đến sự tương tác lâu dài giữa khách hàng và sản phẩm.

**1.3 Cấu trúc của mã QR**

Các khung lưới caro đen và trắng ở cái nhìn đầu tiên có vẻ giống như trò chơi đoán ô chữ được tạo ra random. Nhưng nếu ta nhìn kĩ, có có 1 cấu trúc được xác định. Để các máy scan có thể nhận ra mã QR, nó phải ở dạng hình vuông. Và một số thành phần dưới đây sẽ đảm bảo cho việc thông tin được đọc một cách chính xác.



*Hình 2: Cấu trúc của mã QR*

## 1.3.1 Cell (Ngôn ngữ)

Trước hết, trong mã QR có chứa nhiều ô hoa văn đen trắng, thực tế, các ô đen trắng này chứa các đoạn mã nhị phân. Các ô (cell) trắng đen này lần lượt mang giá trị 0 và 1, tập hợp các cell chính là các thông tin được lưu trữ vào mã QR.

## 1.3.2 Hoa văn định vị

Ở bốn góc của mã QR bố trí các ô vuông gọi là hoa văn định vị. Nhờ vào hoa văn định vị này, camerra có thể xác định được phạm vi mã QR cũng như đọc được thông tin ngay cả trong trường hợp mã QR bị biến dạng, nhờ đó ta có thể quét được mã QR một cách nhanh chóng ở bất kỳ góc độ nào.

Thêm vào đó, hoa văn hình vuông được sử dụng dể xác định phạm vi mã QR ngăn cách với các ký tự, hình vẽ xung quanh nó. Đội phát triển mã QR đã phải khảo sát hơn 5000 trang tờ rơi, bao bì, poster, v.v. và thấy rằng hoa văn này là loại hoa văn có tỉ lệ sử dụng thấp nhất. Cả sự sắp xếp hoa văn cùng với tỉ lệ kích thước cũng là kết quả của việc thống kế nhằm đảm bảo phạm vi code được xác định đúng.

## 1.3.3 Timming pattern (Khung thời gian)

Các ô vuông đen trắng được đặt xen kẽ nhau nhằm giúp cho việc xác định toạ độ của mã QR.

## 1.3.4 Alignment pattern (Ký hiệu căn chỉnh)

Ở vùng phía dưới bên phải của mã QR có một hình vuông chứa hình vuông nhỏ khác bên trong, hoa văn này có tác dụng quan trọng, giúp cho việc điều chỉnh lại những chênh lệch phát sinh do camera bị lệch trong quá trình quét.

## 1.3.5 Thông tin format (Chức năng sửa chữa lỗi)

Xung quanh hoa văn định vị là phần chứa thông tin format, quyết dịnh mức độ sửa chữa lỗi của mã QR. Ngoại trừ các phần ② - ⑤, các vùng khác của mã QR là những vùng ta có thể thiết kế. Bên cạnh đó, để giúp cho việc giữ cân bằng giữa các ô đen và trắng trên mã QR, chức năng Mask được thiết lập. Nhờ vào đó, chúng ta không bao giờ thấy QR code chứa toàn các ô đen. Các thông tin lưu trong mã QR được đảm bảo vẫn giữ toàn vẹn nhưng màu sắc của các ô đen trắng có thể đảo ngược để đảm bảo sự cân bằng.

# **1.4 Khả năng lưu trữ và sửa lỗi**

## 1.4.1 Khả năng lưu trữ

Khả năng lưu trữ của mã mã QR có thể nhập tối đa 7.089 chữ số hoặc 4.296 ký tự, trong đó bao gồm dấu câu và ký tự đặc biệt, từ và cụm từ, ký tự chữ Kanji và Kana. Mã QR rất linh hoạt có thể lưu trữ nhiều loại dữ liệu khác nhau, mỗi mã QR có một phần dành riêng để thông báo cho người đọc loại dữ liệu mà nó lưu giữ. Điều này cho phép các loại dữ liệu khác nhau được mã hóa và giải mã dễ dàng. Khi có thêm dữ liệu được thêm vào Mã QR, kích thước Mã tăng lên và cấu trúc Mã trở nên phức tạp hơn.

Đối với từng loại dữ liệu thì được mã hóa cụ thể số lượng các kí tự như sau:

* Số đơn thuần Tối đa 7.089 ký tự
* Số và chữ cái Tối đa 4.296 ký tự
* Số nhị phân (8 bit) Tối đa 2.953 byte
* Kanji/Kana Tối đa 1.817 ký tự

Những dữ liệu phổ biến mà mã QR lưu trữ như:

* Dữ liệu liên hệ: Là các dữ liệu được lưu trữ trong thẻ liên lạc có thể bao gồm tên người, số điện thoại, địa chỉ email và trang web. Đây là những dữ liệu lý tưởng để đưa vào danh thiếp, cho phép ta có thể dễ dàng chuyển chi tiết từ thẻ vào thiết bị kỹ thuật số của mình.
* Dữ liệu lịch: Là những cuộc hẹn của ta với đối tác và khách hàng, có thể được lưu trữ trực tiếp vào nhật ký và khi được quét nó sẽ hiển thị để thông báo nhắc nhở.
* URL: Đây là một URL mà khi được quét bởi trình duyệt điện thoại thông minh để quét. Ví dụ, “tem công nghệ chống giả” được dán trên sản phẩm, nó sẽ đưa người dùng đến một trang web của doanh nghiệp. Tại trang web, ta có thể tham khảo các sản phẩm – dịch vụ cũng như tìm hiểu các thông tin quan trọng về doanh nghiệp.
* Địa chỉ Email: Thông thường sẽ là một email với chủ đề được xác định trước, người nhận sẵn sàng cho người dùng quét để thêm nội dung và gửi để thông báo cho khách hàng những thông tin liên quan hoặc những vấn đề mà khách hàng muốn biết.
* Số điện thoại: Số điện thoại này là của doanh nghiệp sản xuất và số điện thoại của khách hàng. Khi người tiêu dùng quý xuất nguồn gốc sản phẩm sẽ thấy hiển thị các thông tin về sản phẩm và doanh nghiệp, trong đó có số điện thoại. Đồng thời khi truy xuất thì số điện thoại của quý khách cũng sẽ được lưu lại trên hệ thống, giúp nhà sản xuất thống kê được lượt truy xuất và ai là người truy xuất.
* Tin nhắn điện thoại SMS: Một SMS mới có thể được mở sẵn để gửi cho người nhận. Ví dụ cụ thể nhất là khi người nhận truy xuất nguồn gốc sản phẩm bằng cách nhắn tin theo cú pháp được hướng dẫn chi tiết trên tem.
* Văn bản thuần túy: Với khả năng lưu trữ của mã QR**,**văn bản thuần túy của mã có thể lưu trữ tối đa là 7.089 ký tự dữ liệu số hoặc 4.296 ký tự chữ số trong mã QR tiêu chuẩn, mã QR có thể chứa tối đa 35 ký tự số và mã iQR với 40.000 ký tự.
* Vị trí địa lý: Mã QR có thể chứa liên kết đến một vị trí mà khi được quét, người đọc có thể sử dụng để đưa ra chỉ dẫn về một sự kiện nào đó đang được doanh nghiệp của bạn quảng cáo.

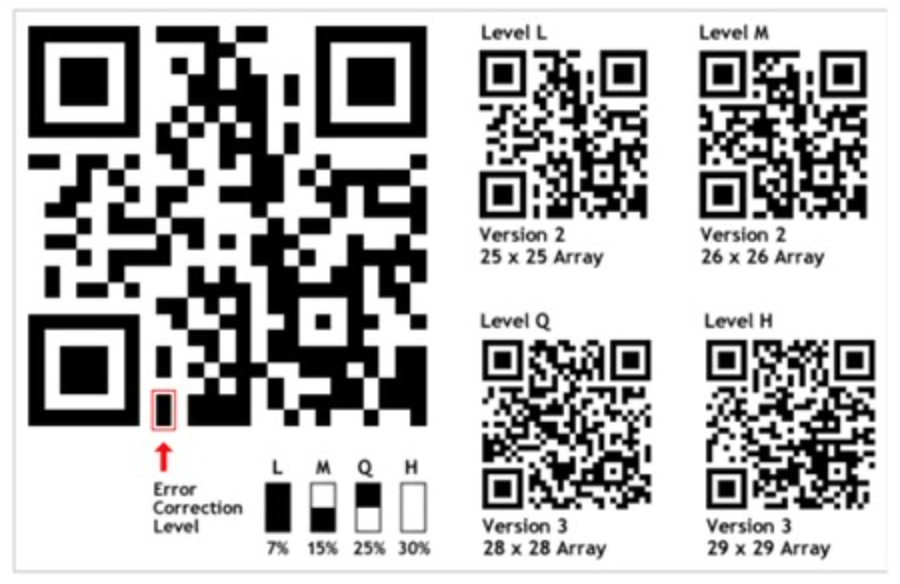
## 1.4.2 Khả năng sửa lỗi

Mã QR có khả năng ứng phó với trường hợp bị bẩn hay rách. Ngay cả khi một bộ phận code bị mất đi nữa, bản thân mã QR có khả năng sửa chữa lỗi, khôi phục data mà nó chứa.

Đó là do khả năng dự phòng và sửa lỗi của mã QR. Vì thật ra thì có đến bốn mức độ khôi phục lỗi trong một mã QR, khi nó được tạo ra. Mỗi mức độ khôi phục sẽ đưa thêm một lượng dữ liệu dự phòng vào trong phần nội dung, để đảm bảo thông tin vẫn được cung cấp đầy đủ, khi hình ảnh bảng mã bị hỏng ở một tỷ lệ nhất định. Bốn cấp độ sẽ là:

* Cấp độ L: cho phép hình ảnh mã hư hại 7%.
* Cấp độ M: cho phép hình ảnh mã hư hại 15%.
* Cấp độ Q: cho phép hình ảnh mã hư hại 25%.
* Cấp độ H: cho phép hình ảnh mã hư hại 30%.

Để biết mã QR đang có cấp độ dự phòng nào, bạn có thể nhìn vào góc trái của hình ảnh.



*Hình 3: Khả năng sửa lỗi qua 4 cấp độ*

Một số vấn đề cần quan tâm về mức độ dự phòng hư hỏng:

* Mức độ dự phòng lỗi càng thấp, hình ảnh càng đơn giản, và bạn có thể in nó với kích thước nhỏ, mà vẫn rõ nét và dễ đọc.
* Mức độ dự phòng lỗi càng cao, hình ảnh càng phức tạp, và gây khó cho chương trình quét.

Bạn chỉ nên chọn mức độ L hoặc M để có sự cân bằng giữa việc dự phòng lỗi, và sự rắc rối của mã QR được tạo ra.

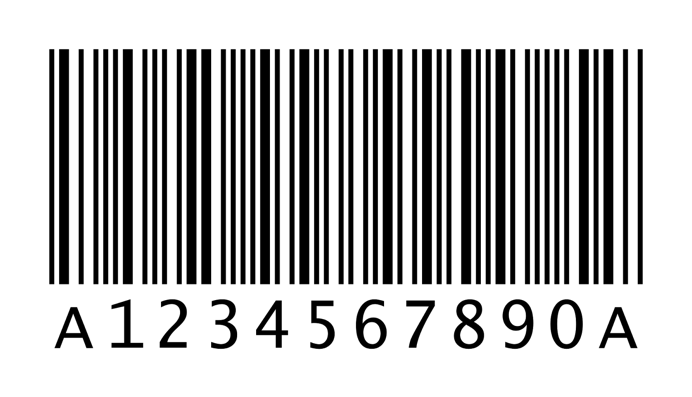
Chính vì có sự khác biệt về cách chọn lựa mức độ dự phòng, nên các chương trình tạo mã QR khác nhau sẽ có thể tạo ra các hình ảnh mã QR hoàn toàn khác nhau, dù chứa cùng một nội dung giống hệt.

# **1.5 So sánh QRCode và BarCode**

## 1.5.1 BarCode là gì?

Ai đã mua đồ ở các cửa hàng tiện lợi thì đều đã nhìn thấy BarCode, tuy nhiên không biết họ có nhận ra là các mã BarCode đó đều hoàn toàn khác nhau. 1 BarCode (mã vạch) là 1 tấm ảnh hiển thị thông tin mà máy quét có thể đọc được về thông tin của sản phẩm được ghi vào nó, là 1 tập hợp các đường thẳng đen song song với chiều rộng (width) khác nhau, hình thành 1 hình chữ nhật nhỏ, đính lên 1 góc của sản phẩm.

Mã BarCode trên sản phẩm chứa thông tin về nhà sản xuất, loại sản phẩm, giá, .. những thông tin mà có thể đọc được bằng những máy đọc chuyên dụng. Bởi vì nó chứa thông tin chỉ theo 1 chiều ngang (horizontal direction), nên nó được gọi là chứa thông tin 1D (1-dimensional).

****

*Hình 4: Ví dụ về mã BarCode*

## 1.5.2 Sự khác biệt giữa QRCode và BarCode

**1.5.2.1 Cách hiển thị**

Có lẽ là sự khác biệt lớn nhất mà ai nhìn vào cũng thấy rõ, chính là hình dáng của chúng. QRCode chứa các mảng màu đen trắng hình thành trong 1 hình vuông, trong khi BarCode chứa các sọc đen trắng song song trong 1 hình chữ nhật.

**1.5.2.2 Khả năng lưu trữ thông tin**

Trong khi BarCode chỉ giữ được thông tin theo chiều ngang (horizontal direction), QRCode có thể giữ thông tin cả chiều ngang (horizontal direction) và chiều dọc (vertical direction). Với sự khác biệt về cấu trúc này, QRCode có thể lưu trữ thông tin nhiều gấp hàng trăm lần (hundreds of times) so với BarCode, do đó nó có thể lưu trữ thông tin tốt hơn trong 1 khoảng diện tích nhỏ hơn so với BarCode.

**1.5.2.3 Khả năng chịu lỗi**

Đây chính là ưu điểm vượt trội của QRCode so với BarCode. QRCode có khả năng chịu lỗi từ 7-30%. Điều này có nghĩa là gì? Tức là, trong trường hợp QRCode in trên sản phẩm bị bẩn hay trầy xước, trong mức cho phép 7-30%, chúng ta vẫn có thể lấy được thông tin trên đó 1 cách chính xác. Nhờ tính năng chịu lỗi vô cùng lớn này, nhiều công ty đã đưa logo hay hình ảnh của họ vào code để phòng trường hợp có bất kì câu hỏi nào liên quan.

Mặc dù BarCode đã phát triển mạnh trong những thập kỉ vừa qua, nhưng với sự ra đời của QRCode với những tính năng hoàn toàn vượt trội, và với sự phát triển mạnh mẽ của Smart phone, QR đã và đang trở thành xu hướng (trend) trong các sản phẩm và dịch vụ hằng ngày.

# **1.6 Ưu điểm và nhược điểm của mã QR**

## 1.6.1 Ưu điểm

Mã QR cung cấp một số lợi thế so với mã vạch truyền thống, những mã này giúp tăng tính linh hoạt, độ tin cậy và dễ sử dụng. Những lợi thế này giúp giảm chi phí triển khai đã giúp tăng mức độ phổ biến của Mã QR như:

* **Dung lượng cao** – lưu trữ nhiều dữ liệu hơn, cho phép lưu trữ nội dung thực và không chỉ các ID hoặc tham chiếu.
* **Yêu cầu ít không gian hơn** – Lấy cùng một dữ liệu được lưu trữ trên một diện tích bề mặt nhỏ hơn nhiều.
* **Khả năng chống bụi và thiệt hại** – Ngay cả khi bị hư hỏng vẫn có cơ hội chúng vẫn có thể đọc được.
* **Có thể đọc được từ bất kỳ hướng nào** – Quét chúng từ mọi góc độ, người đọc không cần phải căn chỉnh theo hướng của mã.
* **Cấu trúc phụ thêm** – Dữ liệu có thể được chia thành nhiều mã khi quét có thể được kết hợp để tái tạo lại nội dung gốc.

## 1.6.2 Nhược điểm

Những ứng dụng trong tương lai dành cho mã vạch mã QR là rất lớn, nhưng mã vạch này cũng có những hạn chế nhất định như sau:

Mã QR đòi hỏi phải có phần mềm đọc mã trên điện thoại, và nếu mã này xuất hiện ngoài trời như trên billboard, bến xe… có thể những chướng ngại vật khác sẽ hạn chế khả năng nhận dạng của máy. Bạn phải có smart phone tích hợp camera mới đọc được mã.

Mã QR cũng không có tính lan truyền (viral) như hình thức nhắn tin SMS…

Hơn nữa, tuy mã QR cực kỳ phổ biến tại Nhật Bản, song nhìn chung loại mã này vẫn còn khá mới mẻ. Thông thường doanh nghiệp phải giải thích và hướng dẫn khách hàng cách sử dụng.

# **1.7 Ứng dụng**

Tiếp sau, chúng ta hãy cùng xem những ứng dụng của mã QR trong các ví dụ thực tế.

**Thanh toán bằng mã QR**

Việc thanh toán bằng mã QR ngày càng trở nên phổ biến. Ta có thể dễ dàng thanh toán bằng cách dùng app chuyên dụng để đọc mã QR chuyên dụng đặt ở quẩy thanh toán, hoặc để nhân viên cửa hàng quét mã QR trên điện thoại của mình. Đặc biệt, ở Trung Quốc và Hàn Quốc, việc thanh toán bằng mã QR đã phổ biến đến mức dù ở bất cứ đâu, bách hoá, nhà hàng cao cấp hay cửa hàng ven đường, ta đều không cần dùng đến tiền mặt nữa.

**Lưu thông tin account LINE?Twitter dưới dạng mã QR**

Khi follow bạn bè trên SNS, nhiều người thường tìm kiếm bạn bè bằng ID hay tên gọi. Tuy nhiên, việc nhập chuỗi ký tự dài thật phiền phức đúng không nào? Nhờ vào mã QR, vấn đề trên đã được giải quyết. Bằng thao tác quét mã QR, ta có thể tìm được account LINE/Twitter của bạn bè một cách dễ dàng. Chức năng này đặc biệt tiện lợi khi trao đổi thông tin, làm quen với những người bạn mới gặp.

**Quản lý người ra vào**

Trong một số sự kiện hoặc buổi hoà nhạc, trận đấu bóng, mã QR được in trên vé, giúp cho việc soát vé. Sau khi đặt vé trên Internet, người tham gia sẽ nhận được vé điện tử chứa mã QR, sau đó, khi đến sự kiện, chỉ cần quét mã QR nhận được, ta đã có thể vào trong khán đài. Tương tự, ở các sân bay, cửa soát vé tàu, ta cũng có thể thấy mã QR được áp dụng dưới dạng Cửa soát vé điện tử.

**Quản lý quy trình, quản lý kho trong ngành sản xuất.**

Đây là một ứng dụng ít được biết đến hơn của mã QR, tuy nhiên, ban đầu mục đích mã QR được tạo ra là nhằm phục vụ cho ngành sản xuất. Ngoài tác dụng quản lý số lượng lưu kho, bằng việc gắn cho mỗi sản phẩm một mã QR riêng biệt, ta có thể dễ dàng truy xuất nguồn gốc, quy trình sản xuất của sản phẩm đó. Ngoài ra, mã QR cũng được ứng dụng trong việc quản lý xuất nhập kho, quản lý số lượng bán ra, số lượng tồn kho trong các cửa hàng.

**Trong tương lai, mã vạch QR Code có thể được sử dụng trong các lĩnh vực như:**

* Lưu trữ URL: Điện thoại chỉ việc đọc mã QR để lấy URL, sau đó tự động mở trình duyệt.
* Sử dụng tại các bến xe bus: người sử dụng khi quét mã QR Code của bến xe bus sẽ biết thông tin về các chuyến xe sắp tới.
* Sử dụng tại bảo tàng: người sửa dụng chỉ cần quét mã QR Code đặt cạnh vật trưng bày là biết được thông tin chi tiết và cập nhật về đồ vật đó/
* Sử dụng để mua hàng ở bất kỳ đâu: người sử dụng khi đi tàu điện ngầm, xe bus… nếu thấy thích mặt hàng đang quảng cáo trên đó có thể đặt mua ngay lập tức qua QR Code và Mobile Internet.
* Sử dụng tại siêu thị: người mua có thể quét mã QR Code để biết được hàm lượng dinh dưỡng của đồ ăn cần mua.
* Sử dụng tại các hội thảo: người tham gia hội thảo có thể sử dụng QR Code thay cho Business Card của mình
* Sử dụng với các tờ báo/tạp chí giấy: người đọc có thể quét mã QR Code được in trong tờ báo/tạp chí giấy để truy cập phiên bản online/mobile của tờ báo/tạp chí này.
* Sử dụng tại quán bar/club: để xác định xem ca sỹ, ban nhạc, bài nhạc đang chơi là ai/bài nhạc gì?
* Sử dụng với các món ăn: để biết được công thức và cách chế biến món ăn.
* Sử dụng với đồ vật cá nhân (xe, áo thun…): để cung cấp thông tin về chủ nhân…
* Sử dụng để thu tiền xe ô tô: thay vì việc phải dừng lại ở trạm thu phí, xe ô tô có thể di chuyển qua trạm với tốc độ vừa phải mà vẫn nộp được phí nếu được gắn QR Code trên xe.

# 

# **CHƯƠNG 2: THIẾT KẾ VÀ BẢO MẬT MÃ QR**

# **2.1 Thiết kế mã QR**

# **2.2 Thuật toán mã hoá mã QR**

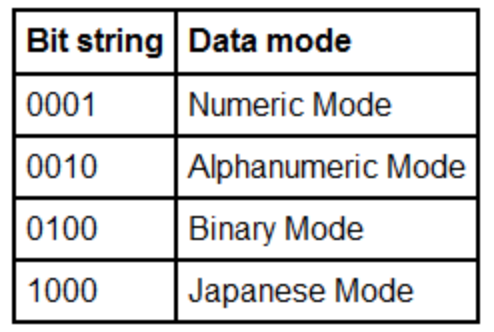
## 2.2.1 Tạo một chuỗi nhị phân

Bước đầu tiên để tạo một mã QR code là tạo một chuỗi nhị phân bao gồm dữ liệu và thông tin về chế độ mã hóa, cũng như độ dài của dữ liệu.

Trong phần hướng dẫn này sẽ hướng dẫn mã hóa chuỗi HELLO WORLD ở QR code phiên bản 1, với mức độ sửa lỗi là Q.

**Bước 1:**Mã hóa Bộ chỉ chế độ (Mode Indicator)

Một bộ chỉ chế độ là một chuỗi 4 bit thể hiện chế độ dữ liệu ta đang sử dụng: kiểu số, kiểu chuỗi, kiểu nhị phân hoặc Nhật ngữ.



*Hình 5: Mode Indicator*

Với chuỗi HELLO WORLD, là kiểu chuỗi, vậy nên Bộ chỉ chế độ sẽ là 0010.

Chuỗi nhị phân: 0010

**Bước 2:** Mã hóa độ dài của dữ liệu

Trong bước này, chúng ta xác định có bao nhiêu ký tự trong thông điệp, và chuyển độ dài đó thành một số nhị phân.

Với chuỗi HELLO WORLD, có 11 ký tự kể cả khoảng trắng. Ta chuyển 11 sang nhị phân, được 1011.

Xem danh sách ở dưới. Khi mã hóa độ dài của dữ liệu, chúng ta mã hóa nó sử dụng một số lượng bit đặc tả. Như đã nói ở đầu, chúng ta sử dụng QR code phiên bản 1, vậy nên chúng ta cần sử dụng 9 bit để mã hóa độ dài dữ liệu. Ta thêm số 0 vào đầu chuỗi cho đên khi đủ 9 bit: 000001011.

**Phiên bản 1 tới 9**

* Kiểu số: 10 bits
* Kiểu chuỗi: 9 bits
* Kiểu nhị phân: 8 bits
* Kiểu Nhật ngữ: 8 bits

**Phiên bản 10 tới 26**

* Kiểu số: 12 bits
* Kiểu chuỗi: 11 bits
* Kiểu nhị phân: 16
* Kiểu Nhật ngữ: 10 bits

**Phiên bản 27 tới 40**

* Kiểu số: 14 bits
* Kiểu chuỗi: 13 bits
* Kiểu nhị phân: 16 bits
* Kiểu Nhật ngữ: 12 bits

Tới đây chuỗi nhị phân chúng ta là: 0010 000001011.

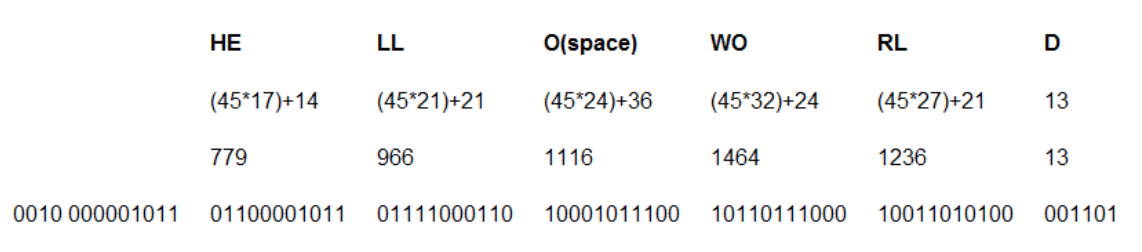
**Bước 3:** Mã hóa dữ liệu

Để mã hóa chuỗi ký tự, ta chia chuỗi thành các cặp ký tự: HE,LL,O ,WO,RL,D.

Với mỗi cặp ký tự, ta lấy giá trị mã ASCII của ký tự đầu nhân với 45, sau đó cộng với giá trị mã ASCII của ký tự sau. Sau đó chuyển kết quả thu được thành một chuỗi nhị phân 11 bit.

Đối với các ký tự không có cặp, tao lấy giá trị mã ASCII của nó rồi chuyển thành chuỗi nhị phân 6 bit.

Đối với các chế độ khác: kiểu số, kiểu nhị phân, kiểu Nhật ngữ, tao sử dụng phương thức khác để mã hóa dữ liệu.



*Hình 6: Kiểu dữ liệu hóa dữ liệu*

**Bước 4:** Hoàn thành các bit

Với chuỗi bit thu được ở trên, ta phải chắc chắn rằng nó có đủ độ dài. Điều này phụ thuộc vào việc có bao nhiêu bit chúng ra cần để tạo phiên bản và sửa lỗi.

Với QR code phiên bản 1, sửa lỗi mức độ Q. Chúng ta phải tạo một chuỗi 104 bit. Tham khảo thêm: [version capacity table](http://www.denso-wave.com/qrcode/vertable1-e.html). Nếu chuỗi bit nhỏ ít hơn 104, ta phải thêm 4 số 0 vào cuối chuỗi. Nếu thêm 4 số 0 được một chuỗi nhiều hơn 104, ta chỉ cần thêm số 0 cho tới khi nào đủ 104 bit.

Chuỗi của chúng ta có 59 bit, ta thêm 4 số 0 vào cuối chuỗi. (Còn nếu chuỗi có 102 bit thì ta thêm 2 số 0).

Chuỗi của ta bây giờ là:

0010 000001011 01100001011 01111000110 10001011100 10110111000 10011010100 001101 **0000**

**Bước 5:** Giới hạn chuỗi thành các chuỗi con 8 bit, sau đó thêm các số 0 nếu cần.

Bước này ta chia chuỗi thành các nhóm 8 bit.

Ở chuỗi con cuối cùng, nếu chưa đủ 8 bit thì ta thêm vào sau đó các số 0 cho tới khi đủ. Chuỗi của chúng ta thêm 2 số 0 vào sau chuỗi con cuối cùng.

00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 010000**00**

**Bước 6:  Thêm các từ vào cuối nếu chuỗi quá ngắn**

**Nếu chuỗi bit của chúng ta vẫn chưa đủ dài, có hai chuỗi con đặc biệt:**11101100 và 00010001. Ta thêm thay phiên hai chuỗi con này vào chuối chuỗi của chúng ta.

Chuỗi của chúng ta chỉ có 10 chuỗi con, cần phải thêm vào 3 chuỗi con nữa. Bắt đầu thêm 11101100, sau đó thêm 00010001, rồi tới 11101100. Nếu cần nhiều hơn thì cứ thêm lần lượt như vậy cho tới khi đủ số chuỗi con.

Chuỗi của chúng ta bây giờ:

00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000 **11101100 00010001 11101100**

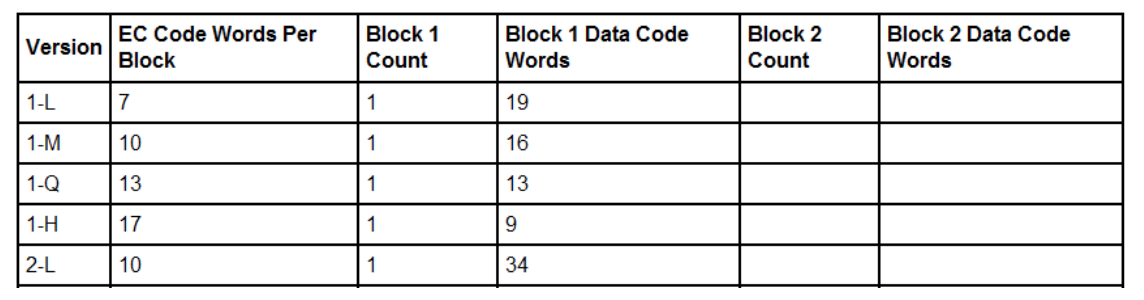
## 2.2.2 Tạo bộ mã sửa lỗi

QR code bao gồm bộ mã sửa lỗi. Các khối dữ liệu dự phòng đảm bảo rằng mã QR code vẫn được đọc cho dù có một phần không đọc được. Tạo mã này rất phức tạp, vì thế nên sau đây sẽ hướng dẫn một cách đơn giản.

QR code sử dụng bộ sửa lỗi Reed-Solomon.

**Bước 1:** Tìm ra có bao nhiêu mã sửa lỗi cần được tạo

Bước đầu tiên để tạo mã sửa lỗi là xác định có bao nhiêu từ để tạo cho phiên bản QR và mức độ sửa lỗi. Xem [bảng sửa lỗi](http://www.thonky.com/qr-code-tutorial/error-correction-table/) để biết thêm thông tin chi tiết



*Hình 7: Bảng sửa lỗi*

Xem bảng ở trên, chúng ta cần 13 mã sửa lỗi.

**Bước 2:** Tạo một thông điệp đa thức

Chúng ta chuyển 13 khối dữ liệu ở trên thành dạng thập phân:

00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000 11101100 00010001 11101100

Ta được:  
32, 91, 11, 120, 209, 114, 220, 77, 67, 64, 236, 17, 236

Các số này sẽ là hệ số của đa thức. Ta có:

*32x25 + 91x24 + 11x23 + 120x22 + 209x21 + 114x20 + 220x19 + 77x18 + 67x17 + 64x16 + 236x15 + 17x14 + 236x13*

Lũy thừa của số hạng đầu tiên được tính:

Số khối dữ liệu + số mã sửa lỗi -1

Ở đây là 13+13-1=25. Vậy nên số hạng đầu tiên là *32x25 .*

**Bước 3:** Tạo bộ tạo đa thức

Tiếp theo ta sẽ tạo một bộ tạo đa thức. Ta chia thông điệp đa thức bởi bộ tạo này để tạo mã sửa lỗi.

Các bộ tạo đa thức đến từ một cái gì đó gọi là một trường hữu hạn , cũng được biết đến như là một trường Galois. Các mã QR sử dụng một trường Galois có 256 yếu tố, có nghĩa là, mục đích của chúng ta, rằng những con số mà chúng ta sẽ được giao dịch với sẽ luôn luôn tối đa là 255 và ít nhất là 0.

Để thực hiện việc phân chia đa thức, chúng ta sẽ được chuyển đổi qua lại giữa các ký hiệu α (alpha) và các số nguyên. Alpha và các giá trị số nguyên từ bảng log và antilog.

Bộ tạo đa thức luôn luôn có dạng *(x – α) (x – α 2) … (x – α t), , t* tương đương với số lượng mã sửa lỗi cần thiết trừ đi 1.Chúng ta cần 13 mã sửa lỗi, do đó, t trong trường hợp này là 12.

Ta thức hiện triển khai tất cả các thành phần (x – α) cho tới khi chúng ta được đa thức. Công việc này thực hiện bằng tay khá phức tạp, sử dụng [generator polynomial tool](http://www.thonky.com/qr-code-tutorial/generator-polynomial-tool/). Ở đây chúng ta tạo 13 mã sửa lỗi:

α0x25 + α74x24 + α152x23 + α176x22 + α100x21 + α86x20 + α100x19 + α106x18 + α104x17 + α130x16 + α218x15 + α206x14 + α140x13+ α78x12

Dưới đây, tôi cho thấy từng bước của việc phân chia đa thức cần thiết để tạo ra mã sửa lỗi. Nếu bạn muốn xem các bước này được thực hiện cho một đa thức tin nhắn khác nhau và số lượng từ ECC, truy cập trang [show division steps](http://www.thonky.com/qr-code-tutorial/show-division-steps/).

Kết quả của bước này ta được mã sửa lỗi:

168 72 22 82 217 54 156 0 46 15 180 122 16

Ta đặt đoạn mã sửa lỗi sau đoạn mã dữ liệu:

32 91 11 120 209 114 220 77 67 64 236 17 236 **168 72 22 82 217 54 156 0 46 15 180 122 16**

**Chuyển qua nhị phân:**

00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000 11101100 00010001 11101100 10101000 01001000 00010110 01010010 11011001 00110110 10011100 00000000 00101110 00001111 10110100 01111010 00010000

## 2.2.3 Chọn mẫu mặt nạ tốt nhất

Bây giờ chúng ta đã mã hóa các dữ liệu, tất cả những gì còn lại là chọn mẫu mặt nạ tốt nhất. Một mẫu mặt nạ thay đổi theo các bit, theo một quy tắc cụ thể. Các mẫu mặt nạ được định nghĩa trong tiêu chuẩn mã QR.

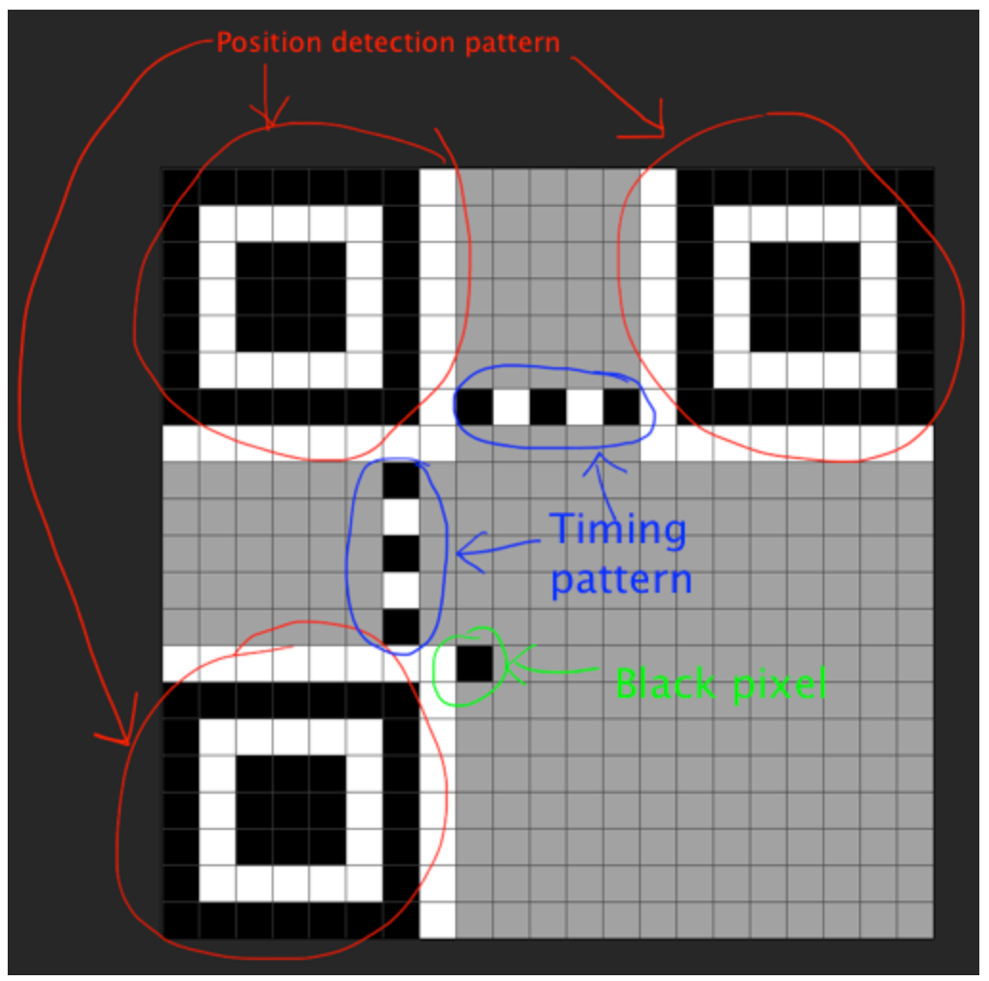
Có 8 mẫu mặt nạ. Xem danh sách các [kiểu mặt nạ QR](http://www.thonky.com/qr-code-tutorial/mask-patterns/) . Sử dụng mô hình mặt nạ cho phép chúng tôi để tạo ra tám mã QR khác nhau và sau đó chọn một trong đó sẽ được dễ dàng nhất cho một đầu đọc QR để quét. Một đặc biệt mã QR có thể có các mẫu nhất định hoặc những đặc điểm mà làm cho nó khó khăn cho thiết bị đọc QR chính xác quét mã QR. Ví dụ, nếu điểm ảnh của cùng một màu sắc xảy ra gần nhau, một đầu đọc QR có thể gặp khó khăn khi đọc chính xác các điểm ảnh.

Mỗi mô hình mặt nạ tạo ra một mã QR code khác nhau. Sau khi chúng ta tạo ra các mã khác nhau QR trong nội bộ, chúng tôi cung cấp cho mỗi một số điểm xử phạt theo các quy tắc quy định trong tiêu chuẩn mã QR. Sau đó, chúng tôi cho ra mã QR có số điểm tốt nhất.

**Bước 1:** Tạo mã QR code:

Với QR code phiên bản 1, ta cần một ma trận 21×21.

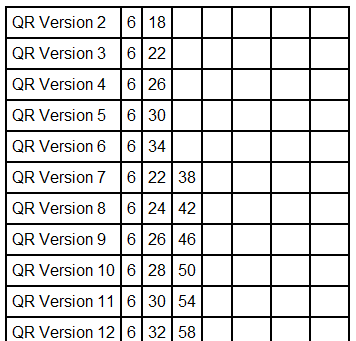
Tất cả các mã QR code đều có một vài mẫu nhận diện chắc chắc phải đc thể hiện. Ở hình bên dưới, có ba vị trí nhận diện được đặt ở trên cùng bên trái, trên cùng bên phải và dưới cùng bên trái của ma trận. Các mã QR code còn có một pixel đen ở bên phải của vị trí nhận diện bên dưới cùng bên trái. Ngoài ra còn có mẫu timing.

****

*Hình 8: Hình nhận diện QR Code*

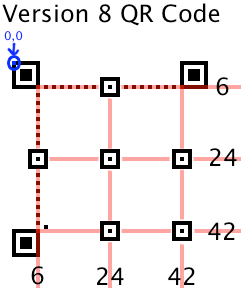
**Vị trí điều chỉnh mô hình**

Từ ví dụ này là chỉ mã hóa một phiên bản 1 mã QR, chúng tôi không cần thêm bất kỳ mô hình điều chỉnh vị trí. Tuy nhiên, nếu chúng ta đã mã hóa một mã QR phiên bản 2 hoặc lớn hơn, chúng tôi sẽ thêm các mẫu điều chỉnh vị trí ma trận. Bảng dưới đây liệt kê các tọa độ của nơi để đặt các mô hình điều chỉnh vị trí. Xem ví dụ ở bảng dưới đây.

****

*Hình 9: Bảng liệt kê tọa độ của nơi để đặt các mô hình điều chỉnh vị trí*

Ví dụ, hình ảnh dưới đây cho thấy các mô hình điều chỉnh vị trí cho phiên bản mã QR 8. Mỗi mô hình điều chỉnh vị trí là một điểm ảnh màu đen bao quanh bởi các điểm ảnh màu trắng được bao quanh bởi các điểm ảnh màu đen. Theo bảng trên, cho phiên bản mã QR 8, vị trí các mô hình điều chỉnh 6, 24, và 42. Như ta thấy trong hình ảnh, mô hình điều chỉnh vị trí được đặt ở tất cả các giao điểm của ba con số này, ngoại trừ đã có lớn hơn mô hình phát hiện vị trí **không đặt mô hình điều chỉnh vị trí trên hàng đầu của các mô hình phát hiện vị**trí trong **góc của mã QR.**

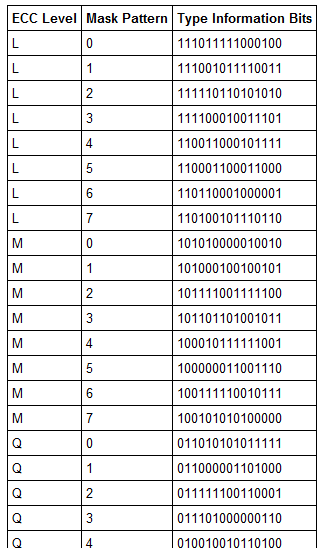
****

*Hình 10: Mô hình điều chỉnh vị trí*

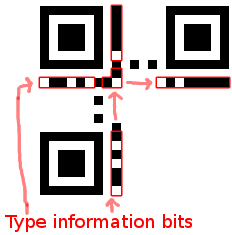
**Thêm thông tin Loại**

Các thông tin về mức độ sửa lỗi và mô hình mặt nạ được mã hóa ở dạng dải với các cạnh của các mẫu phát hiện vị trí.

Bảng sau đây hiển thị các bit thông tin các loại được yêu cầu cho các cấp, sửa lỗi khác nhau và các mẫu mặt nạ.

****

*Hình 11: Bảng hiển thị các Bit thông tin*

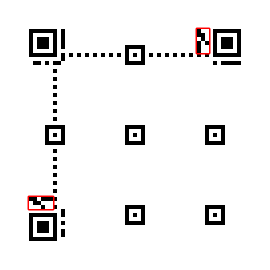


*Hình 12: Hình hiển thị các Bit thông tin*

Trong ví dụ Hello World của chúng tôi, mô hình mặt nạ tốt nhất là 0 (điều này được giải thích dưới đây). Chúng ta đang sử dụng mức độ hiệu chỉnh lỗi Q, vì vậy các chuỗi loại thông tin là 011010101011111. Như trong hình trên, các chuỗi loại thông tin được hiển thị hai lần. Một trong số đó là dưới hai mô hình phát hiện vị trí hàng đầu, bắt đầu từ bên trái. Nó bỏ qua quá khứ mô hình thời gian và phần dữ liệu của lưới. Điều thứ hai là quyền của các mẫu phát hiện vị trí, bắt đầu từ phía dưới, bỏ qua dấu chấm đen, phần dữ liệu, và mô hình thời gian. Cả hai sọc cùng một chuỗi: 011010101011111.

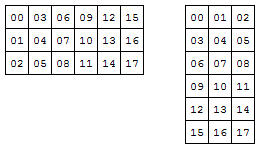
**Thêm thông tin phiên bản**

Nếu mã QR của bạn là phiên bản 7 hoặc lớn hơn, bạn cần phải thêm các bit thông tin phiên bản ma trận. Đây là những được đặt ở bên trái của mô hình trên bên phải phát hiện vị trí và trên các mô hình phát hiện vị trí phía dưới bên trái, như thể hiện trong hình ảnh dưới đây, trong đó có một phiên bản 9 Mã QR.

****

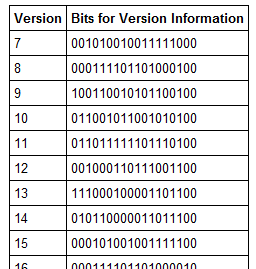
*Hình 13: Hình thêm thông tin phiên bản QR Code*

Bảng sau đây cho thấy một danh sách của các bit thông tin phiên bản cần thiết cho mỗi phiên bản. Những bit được đặt trong cấu hình sau đây trong các vị trí được chỉ định trong hình trên:

****

*Hình 14: Bảng danh sách các Bit thông tin phiên bản*

Trong ví dụ của chúng ta, mã QR là nhỏ hơn so với phiên bản 7, vì vậy chúng ta không cần thêm các bit thông tin phiên bản mã.

****

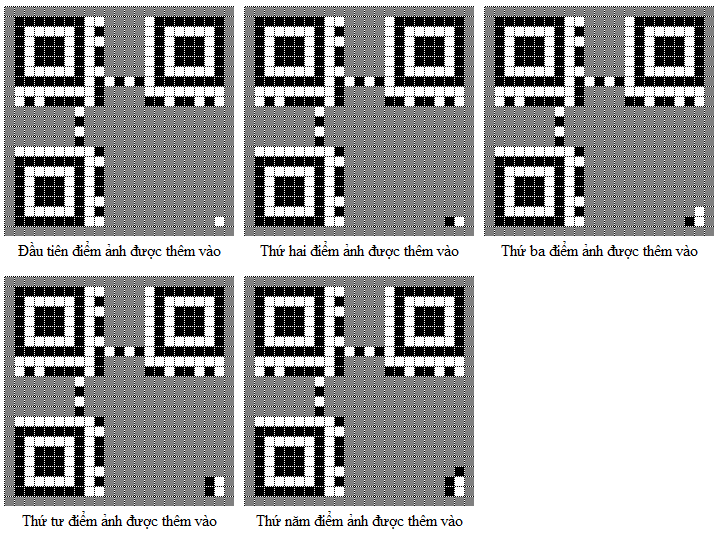
*Hình 15: Bảng Bit tương ứng từng phiên bản*

**Thêm dữ liệu Bits**

Cuối cùng, chúng ta thêm các bit dữ liệu. Trong bước này, chúng ta thay đổi chút tùy thuộc vào mô hình mặt nạ mà chúng ta đang sử dụng.Trước khi thêm chút thực tế, chúng tôi sử dụng các mẫu mặt nạ để quyết định xem có nên thay đổi màu sắc của các bit hiện tại hay không.

**Sắp xếp theo cột từ dưới lên**

Các bit dữ liệu được thêm vào theo một thứ tự cụ thể. Các điểm ảnh đầu tiên được thêm vào phía dưới bên phải của mã QR. Điều thứ hai được đặt ở bên trái đó. Hai tiếp theo được đặt trên đầu những người đầu tiên theo thứ tự. Hai cột điểm ảnh này vẫn tiếp tục đi lên theo cách này, bỏ qua bất kỳ điểm ảnh đã được thiết lập, cho đến khi nó đạt đến hàng đầu của mã QR.

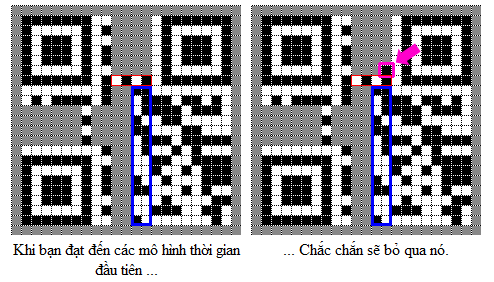
****

*Hình 16: Hình ảnh xuống cột*

Khi bạn đạt đến dưới cùng của lưới, thay đổi hướng một lần nữa, và tiếp tục trong mô hình này giống như đường đi của con rắn.

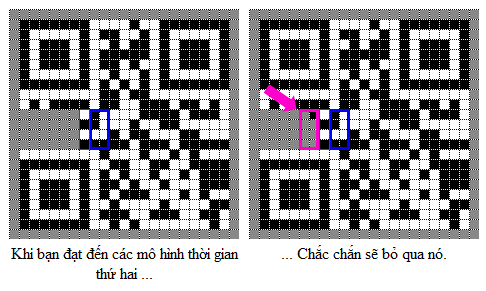
**Chắc chắn bỏ qua mẫu timing**

Khi bạn thêm pixels, hãy chắc chắn để bỏ qua các mô hình thời gian.

****

*Hình 17: Hình ảnh các mẫu timing*

Và các mẫu timing ở chiều dọc:

****

*Hình 18: Hình các mẫu timing ở chiều dọc*

**Hoàn tất mã QR code**

Tiếp tục theo cách này cho đến khi tất cả của các điểm ảnh dữ liệu đã được đặt. Sau đó, các mã QR là hoàn tất. Tuy nhiên, các đặc điểm kỹ thuật mã QR yêu cầu bạn sử dụng các mô hình mặt nạ tốt nhất có thể

Bước 3: Cung cấp mỗi QR code trong 8 mã một số điểm và chọn mã có số điểm tốt nhất

Trước khi chúng ta có thể cho ra mã QR của chúng ta, chúng ta cần thử lần lượt 8 loại của các mẫu mặt nạ để tìm ra một penalty thấp nhất, dựa trên các quy tắc penalty quy định tại các đặc điểm kỹ thuật mã QR.

Có bốn quy tắc xử phạt:

* Quy tắc đầu tiên cung cấp cho các mã QR một hình phạt đối với từng nhóm năm hoặc nhiều hơn điểm ảnh cùng màu trong một hàng.
* Nguyên tắc thứ hai cung cấp cho các mã QR một hình phạt đối với từng khu vực 2×2 điểm ảnh cùng màu mà nó tìm thấy.
* Nguyên tắc thứ ba cung cấp cho các mã QR một hình phạt lớn nếu có mô hình trông tương tự như các mô hình phát hiện vị trí.

Nguyên tắc thứ tư cung cấp cho các mã QR một hình phạt nếu có nhiều hơn một nửa trong số các điểm ảnh tối hoặc ánh sáng

# **2.3 Bảo mật mã QR với thông tin ba lớp sử dụng mã Hamming**

Ngày nay, mã QR được sử dụng rộng rãi trong các lĩnh vực về hậu cần và vận chuyển, bán sản phẩm, và nhiều hoạt động khác các lĩnh vực yêu cầu quản lý thông tin tự động. Tuy nhiên, vì mã QR là một ký hiệu máy có thể đọc được, phương pháp mã hóa làm cho thông tin lưu trữ của nó thiếu bảo mật bảo vệ . Để cải thiện tính bảo mật của mã QR ứng dụng, các học giả trong và ngoài nước đã đề xuất một hàng loạt giải pháp bảo mật.

Dựa trên sơ đồ mật mã truyền thống, Wang và cộng sự đã đề xuất một mã hóa quang học phi tuyến lược đồ dựa trên biến đổi Fresnel bị cắt ngắn theo pha. Mã QR bí mật đã được mã hóa thành một mã giả ngẫu nhiên hình ảnh nhiễu dưới sự kiểm soát của hai khóa riêng, dẫn đến bảo mật cao. Tuy nhiên, chương trình cần đặc biệt dụng cụ quang học khi giải mã, giúp giảm đáng kể giá trị ứng dụng. Chuang và cộng sự đã sử dụng Lagrange

thuật toán nội suy để chia sẻ thông tin bí mật vào một số phần, và sau đó mã hóa nó thành mã QR hai cấp. Huang và cộng sự đề xuất một kế hoạch xác thực hai chiều dựa trên giao thức Needham-Schroeder và cũng được sử dụng Logic Gong-Needham-Yahalom để phân tích vụ chống tấn công khả năng của giao thức xác thực. Krishna và cộng sự đề xuất một kế hoạch chống hàng giả sản phẩm bằng cách sử dụng thuật toán DES - Data Encryption Standard (Tiêu chuẩn mã hóa dữ liệu), đã mã hóa bản mã được mã hóa cùng với thông tin chữ viết thông thường thành hai cấp mã QR. Trong quá trình phục hồi bí mật, tất cả các tờ giấy trong suốt đã trực tiếp XOR để lấy thông tin bí mật. Tuy nhiên, chương trình có khả năng chống nhiễu yếu. Dựa trên sơ đồ che giấu thông tin, Tkachenko và cộng sự đã thiết kế một số nhận dạng mẫu đơn vị và đã chọn mô-đun màu đen của mã QR từ đơn vị theo trình tự xáo trộn bí mật để nhận ra lưu trữ thông tin hai cấp. Tuy nhiên, QR hai cấp mã do chương trình tạo ra có một mở rộng pixel lớn, và thuật toán trích xuất bí mật có yêu cầu cao hơn về độ phân giải của máy. Chiang và cộng sự sử dụng giấy ướt để nhúng thông tin bí mật vào hình ảnh phương tiện để đạt được chiết xuất điểm mù. Để tăng tải bí mật năng lực, Lin và cộng sự đã cải tiến thuật toán để giới hạn trên của số lượng bit nhúng có thể đạt đến dung lượng dữ liệu tối đa của mã QR. Sau đó, Lin đề xuất một ngưỡng cửa (n, n) kế hoạch chia sẻ bí mật. Các lược đồ thu được luồng n bit bằng thuật toán tạo số ngẫu nhiên và thuật toán hash và sau đó giấu nó trong hình ảnh cung cấp thông tin sử dụng giấy ướt để tạo ra QR hai cấp mã, có thể chống lại nhiễu nhiễu do quá trình in. Tuy nhiên, tỷ lệ nhúng của bí mật thông tin không được xem xét đầy đủ và khả năng bí mật là vẫn thấp. Liu và cộng sự đề xuất chia sẻ hình ảnh mở rộng thuật toán dựa trên mã QR, nhưng chia sẻ bị bóp méo và có thể thu hút sự chú ý của những kẻ tấn công tiềm năng.

Để khắc phục những tồn tại, trong đề án này, đề xuất một tính năng mới và hiệu quả nhúng cao chương trình chia sẻ dựa trên mã QR và mã Hamming, hiện thực hóa mục đích của việc nhúng ba lớp thông tin với độ bền cao. Thông tin lớp đầu tiên được mã hóa thành mã QR thông tin công khai và kết quả XOR của tất cả các chia sẻ được sử dụng làm thông tin lớp thứ hai, sau đó quy tắc sửa đổi trong thông tin lớp thứ hai được sử dụng dưới dạng chỉ mục của thông tin lớp thứ ba và mã Hamming được sử dụng để nhúng lớp thông tin thứ ba.

## 2.3.1 Mã Hamming

Mã Hamming là một mã sửa lỗi tuyến tính, được phát minh bởi Richard Wesley Hamming vào năm 1950. Mã Hamming là một mã hoàn chỉnh điển hình và là một mã hiệu quả cao để sửa một lỗi duy nhất.

Mã Hamming là một loại (n, k) khối tuyến tính mã với ma trận kiểm tra H, có cột bao gồm tất cả các vectơ được sắp xếp lại m không phải là tất cả các số 0 và khác từ nhau, trong đó n, k, m cho biết độ dài của mã hamming, bit chẵn lẻ và bit thông tin.

Hàm (-1, - k- 1) mã Hamming có thể được sử dụng để xây dựng một mã stego ( kỹ thuật giấu tin) và nhúng kbits thư vào -1 pixel bằng cách thay đổi nhiều nhất một trong số chúng. Mã Hamming (7, 4) được lấy ví dụ như sau: chúng ta giải thích cách nhúng và trích xuất 3 bit trong tổng số tin nhắn thành 7 pixel. H là ma trận kiểm tra tính chẵn lẻ của mã Hamming (7, 4) có các cột nằm trong tự nhiên thứ tự tăng dần số nhị phân.

(1)

Cho một từ mã 7 bit x = [1001000] và một tin mật 3 bit m = [110]. Tính toán:

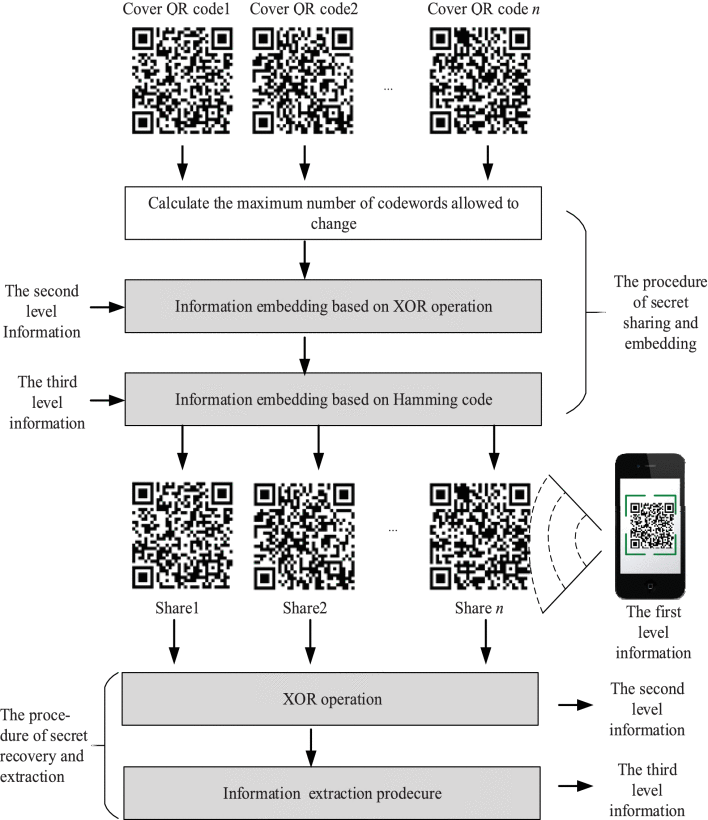
H \* ⊕ m = ⊕ = (2)

Kết quả thu được theo công thức (2) trong đó ⊕ đại diện cho phép toán XOR là một biểu diễn nhị phân của ba, tức là cột thứ ba của ma trận kiểm tra *H*. Thông tin là được nhúng bằng cách thay đổi bit thứ ba của x. Cuối cùng, sự thay đổi luồng bit thông báo là *x*. Khi thông tin bí mật là trích xuất, chỉ tính toán

*H* \* = = (3)

## 2.3.2 Mô phỏng bảo mật

Giả sử thiết kế một (n , n) lược đồ chia sẻ bí mật trong đó có n = 2p . Trong kịch bản đề xuất, có một nhà phân phối bí mật, một nhà soạn nhạc bí mật, n người tham gia. Nhà phân phối bí mật nhúng thông tin lớp thứ hai và thông tin lớp thứ ba vào mã QR của nhà cung cấp dịch vụ để tạo ra một số chia sẻ mã QR và mỗi chia sẻ mã QR có thể được đọc chính xác bởi bộ giải mã mã QR tiêu chuẩn. Khi khôi phục bí mật được thực hiện, cần phải lấy thông tin lớp thứ hai bằng phép toán XOR. Sau đó, thông tin lớp thứ ba được trích xuất.

****

*Hình 19: Hình minh hoạ chương trình đề xuất*

**2.3.2.1 Các giai đoạn chia sẻ bí mật**

Giả định rằng mã QR của nhà cung cấp có dạng Ci ( i = 1 , 2 , … ,2p) khác nhau và có thể được giải mã và đọc bằng bộ giải mã mã QR tiêu chuẩn. Luồng thông tin lớp thứ hai là v và thông tin lớp thứ ba là s.

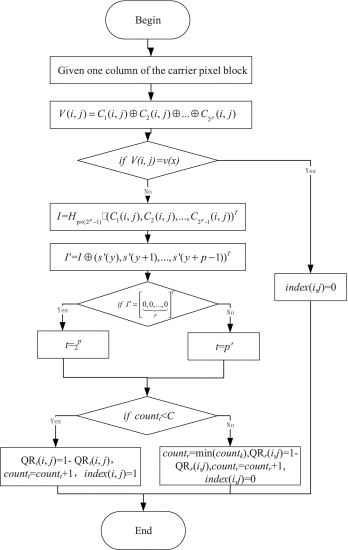
***a.Giai đoạn sơ bộ***

Đề án được đề xuất sử dụng cơ chế sửa lỗi của mã QR để nhúng thông tin. Để biểu thị số lượng từ mã sửa lỗi của mã QR. Theo thuật toán tạo mã QR, có thể biết rằng khả năng sửa lỗi nhỏ hơn một nửa số lượng từ mã sửa lỗi. Do đó, số lượng pixel đảo ngược trong mỗi mã QR của nhà cung cấp dịch vụ do nhúng thông tin lớp thứ hai không thể vượt quá một nửa số từ mã sửa lỗi, có nghĩa là C=⌊E/2⌋×8.

Đề án được đề xuất sử dụng nguyên tắc sửa lỗi của mã Hamming để thực hiện việc nhúng thông tin lớp thứ ba. Do đó, ma trận kiểm tra mã Hamming phải được chia sẻ giữa người phân phối bí mật và người soạn thảo bí mật. Hơn nữa, vì mã Hamming bao gồm các pixel vị trí tương ứng trong 2p – 1 ổ phiếu được sử dụng làm vật mang thông tin lớp thứ ba, cần phải thống nhất về ma trận kiểm tra chẵn lẻ Hp \* (2p -1) của (2p -1, 2p - p -1) mã Hamming trước.

***b.Giai đoạn che giấu***

Quá trình xử lý chương trình của chúng ta cho một pixel Ck ( i , j ) trong hình ảnh sóng mang được hiển thị trong Hình 3 .

****

*Hình 20: Lưu đồ của quá trình trên Ck (i, j)*

Thuật toán chi tiết được cung cấp trong thuật toán 1: Thuật toán che giấu thông tin.

Hình ảnh mã QR của nhà cung cấp dịch vụ  có kích thước m\*n , thông tin lớp thứ hai v , thông tin lớp thứ ba s.

Chỉ số ma trận, chia sẻ QR1 , QR2 ,…,.

Xây dựng khối pixel mạng  và không lấp đầy luồng thông tin lớp thứ ba s để có được luồng thông tin lớp thứ ba mới s′ thỏa mãn *độ dài* (s′) mod p = 0 . Cho i = 1, j = 1, x = 1, y = 1, *countk* = 0 (k = 1, 2,…, 2p ), QRk = Ck , nơi i và j tương ứng đại diện cho hàng và cột của mã QR của nhà cung cấp dịch vụ. x và y tương ứng đánh dấu các vị trí pixel của luồng thông tin lớp thứ hai và luồng thông tin lớp thứ ba, và *countk* ( k = 1, 2, … ,2p) đại diện cho số lượng pixel được thay đổi trong mã QR của nhà cung cấp thứ *k*.

Nếu Ck (i, j) (k = 1, 2, … ,2p ) thuộc về các mẫu chức năng của mã QR, j = j + 1 và chuyển đến *Bước 10*. Nếu không, hãy tính toán kết quả từng phần tử cột của khối pixel V(i, j) = C1(i, j) ⊕ C2( i , j ) ⊕ … ⊕ (i, j).

Nếu V(i, j) = v(x) , để index(i, j) = 0, j = j + 1 , x = x + 1 và chuyển đến *Bước 10* . Nếu không, hãy chuyển sang *Bước 4* .

Tính toán ma trận I = Hp \* (2p -1) \* (C1( i , j ) ,C2( i , j ),...,.

Sử dụng nguyên tắc sửa lỗi của mã Hamming, hãy tính . Nếu  , chuyển đến *Bước 6* . Nếu không, hãy chuyển sang *Bước 7* .

Nếu thì , index(i, j) = 1, y = y + p, j = j + 1 , x = x + 1 , chuyển đến *Bước 8* . Nếu > C, countT = min(countk) (k = 1, 2, … ,2p), QRT(i, j) = 1 – QRT(i, j), countT = countT + 1, index(i, j) = 0, j = j + 1, x = x + 1, và chuyển sang bước 9.

Tìm cùng một cột trong  như I′ và để p′ biểu thị số cột tương ứng. Nếu < C, (i, j) = 1 - (i, j), và chuyển đến *Bước 8* . Nếu (k = 1, 2, … , 2p ), và chuyển đến *Bước 9* .

Nếu y > length(s ′), sau đó index(i, j) = null, việc nhúng thông tin lớp thứ ba đã hoàn thành sau đó chuyển đến *Bước 12*. Nếu không, hãy chuyển đến *Bước 9* .

Nếu x > length (v), việc nhúng thông tin lớp thứ hai đã hoàn thành sau đó chuyển đến *Bước 12*. Nếu không, hãy chuyển đến *Bước 10*.

Nếu j < n , sau đó chuyển sang *Bước 2* . Nếu không, hãy chuyển sang *Bước 11*

Nếu i < m , sau đó chuyển sang *Bước 2* . Nếu không, hãy chuyển đến *Bước 12* .

Thuật toán kết thúc.

Cuối cùng, nhà phân phối bí mật phân phối số cổ phần đã tạo cho những người tham gia. Điều đáng chú ý là thuật toán hạn chế vị trí nhúng của thông tin lớp thứ ba để đảm bảo tính dễ đọc của các chia sẻ, điều này được thể hiện đầy đủ trong thuật toán Bước 6 và 7. Cá nhân tham gia có thể giải mã thông tin công khai của các chia sẻ thông qua QR tiêu chuẩn trình đọc mã nhưng không thể lấy thông tin lớp thứ hai và thông tin lớp thứ ba.

**2.3.2.2 Thủ tục tiết lộ bí mật**

Trong sơ đồ được đề xuất, tất cả các cổ phiếu được yêu cầu thực hiện một thao tác XOR đơn giản xếp chồng để có được thông tin lớp thứ hai. Cuối cùng, ma trận được nhân lên để trích xuất thông tin lớp thứ ba. Các bước cụ thể như sau:

**a.Phục hồi thông tin lớp thứ hai**

#### Thuật toán 2 Thuật toán khôi phục thông tin lớp thứ hai

Các chia sẻ QR 1 , QR 2 ,…, QRp2 có kích thước m×n , số bit trong độ dài thông tin lớp length(v).

Thông tin lớp thứ hai v.

Xây dựng một khối pixel   
 Để i = 1, j = 1, v = {}, count = o, nơi i và j đại diện cho hàng và cột của mã QR của nhà cung cấp dịch vụ tương ứng. v được sử dụng để lưu trữ thông tin lớp thứ hai được trích xuất và số đếm được sử dụng để đếm số bit trong thông tin lớp thứ hai.

Nếu QR*k* (i, j) (k = 1 , 2 , … ,2p ) thuộc về các mẫu chức năng của mã QR, hãy chuyển đến *Bước 6* . Nếu không, hãy chuyển sang *Bước 4* .

Nếu count , chuyển đến *Bước 5* . Nếu không, việc trích xuất thông tin lớp thứ hai đã hoàn tất và chuyển sang *Bước 8* .

Đặt count = count + 1 và tính toán nơi "∪” để trình bày sự hợp nhất của các tập hợp.

j = j + 1 . Nếu j < n , sau đó chuyển sang *Bước 3* . Nếu không, hãy chuyển sang *Bước 7* .

i= i + 1. Nếu i < m , sau đó chuyển sang *Bước 3*. Nếu không, hãy chuyển sang *Bước 8*.

Thuật toán kết thúc.

**b.Trích xuất thông tin lớp thứ ba**

**Thuật toán 3 Thuật toán trích xuất thông tin lớp thứ ba**

Các chia sẻ , ,…,  có kích thước m×n , Chỉ số ma trận.

Thông tin lớp thứ ba S .

Xây dựng một khối pixel

Để cho i = 1 , j = 1 , s = {}, nơi i và j tương ứng, đại diện cho hàng và cột của mã QR của nhà cung cấp dịch vụ. s được sử dụng để lưu trữ thông tin lớp thứ ba đã trích xuất.

Nếu index(i, j) = 0, thì chuyển sang *Bước 5* . Nếu *index*(*i, j*) = 1, hãy chuyển sang *Bước 4* .

, , nơi “” trình bày sự hợp nhất của các tập hợp.

j = j + 1 . Nếu j < n , sau đó chuyển sang *Bước 3* . Nếu không, hãy chuyển sang *Bước 7.*

i= i + 1 . Nếu i < m , sau đó chuyển sang *Bước 3* . Nếu không, hãy chuyển sang *Bước 7*.

Thuật toán kết thúc.

**2.3.3 Bằng chứng bảo mật**

Vì thông tin lớp đầu tiên của các lượt chia sẻ giống với thông tin được lưu trữ bởi mã QR của nhà cung cấp dịch vụ và có thể thu được cả phiên bản và mức sửa lỗi, nên đối thủ có thể tạo chính xác mã QR của nhà cung cấp dịch vụ gốc. Do đó, có thể dễ dàng tính toán sự khác biệt bằng cách so sánh mã QR ban đầu của nhà cung cấp dịch vụ với các cổ phiếu. Mặc dù vậy, nó không thể tái tạo lại bí mật từ thông tin một mình.

#### Định lý 1:

Xác suất thành công để suy ra lớp thứ hai và lớp thứ ba thông tin dựa trên một lượt chia sẻ riêng lẻ sẽ bị giới hạn bởi  và \* tương ứng.

#### Bằng chứng:

Khi đối thủ có và chỉ có một phần riêng lẻ St , có thể áp dụng cho đối thủ một chiến lược bạo lực giống như một cuộc tấn công từ điển. Để choS1⊕S2⊕…⊕ Si -1 ⊕Si+1⊕…⊕Sn = A, từ S1⊕S2⊕…⊕Sn = S ; do đó, đối với mỗi pixel trong St , các St(i, j ) = S(i, j) ⊕ A (i, j) tồn tại, có nghĩa là một pixel bí mật có thể được giải mã chính xác với xác suất 0,5 dựa trên pixel tương ứng trong lượt chia sẻ có sẵn. Điều này giả định rằng độ dài của thông tin lớp thứ hai l1 = *length*(v) có sẵn cho kẻ thù. Do đó, xác suất thành công để suy ra thông tin lớp thứ hai sẽ bị giới hạn bởi .

Vì thực tế là kẻ thù có thể có được độ dài của thông tin lớp thứ 3 l2= length(s) và độ dài của mã Hamming, là 2p -1, kẻ thù có thể suy ra ma trận đồng hành và một mã Hamming tương ứng bằng cách  và tương ứng. Trong khi đó, rất khó để có được vị trí nhúng, dẫn đến1/Cl2m×n xác suất để đoán chính xác các vị trí. Do đó, khả năng thành công của cuộc tấn công thông tin lớp thứ ba này sẽ bị giới hạn bởi \*

#### Bằng chứng:

Khi đối thủ có và chỉ có một phần riêng lẻ St , có thể áp dụng cho đối thủ một chiến lược bạo lực giống như một cuộc tấn công từ điển. Để choS1⊕S2⊕…⊕Si - 1⊕Si + 1⊕…⊕Sn= A , từ S1⊕S2⊕…⊕Sn= S ; do đó, đối với mỗi pixel trong St , các St(i, j) = S(i, j) ⊕ A(i, j) tồn tại, có nghĩa là một pixel bí mật có thể được giải mã chính xác với xác suất 0,5 dựa trên pixel tương ứng trong lượt chia sẻ có sẵn. Điều này giả định rằng độ dài của thông tin lớp thứ hai l1= length(v) có sẵn cho kẻ thù. Do đó, xác suất thành công để suy ra thông tin lớp thứ hai sẽ bị giới hạn bởi.

Vì thực tế là kẻ thù có thể có được độ dài của thông tin lớp thứ ba l2 = length(s) và độ dài của mã Hamming, là 2p - 1, kẻ thù có thể suy ra ma trận đồng hành và một mã Hamming tương ứng bằng cách 1/(2p−1)! và 1/2n-2 tương ứng. Trong khi đó, rất khó để có được vị trí nhúng, dẫn đến xác suất để đoán chính xác các vị trí. Do đó, khả năng thành công của cuộc tấn công thông tin lớp thứ ba này sẽ bị giới hạn bởi \*

#### Định lý 2:

Xác suất thành công để suy ra thông tin lớp thứ hai và lớp thứ ba dựa trên n – 1 cổ phiếu sẽ bị ràng buộc bởi và + tương ứng.

#### Bằng chứng:

Có thể tưởng tượng rằng n -1 người tham gia có thể thông đồng và kết hợp thông tin từ những chia sẻ của họ với nhau để cố gắng tìm S , được gọi là tấn công thông đồng. Để cho S1⊕S2⊕…⊕Sn-1 = B ; do đó, Sn(i, j) = S(i, j) ⊕ B (i, j) tồn tại, tương tự như phân tích trong Định lý 1 . Như trên, xác suất để suy ra thông tin lớp thứ hai sẽ bị giới hạn bởi .

Bởi vì kẻ thù đã n - 1 cổ phiếu, có nhiều nhất một cổ phiếu không thể tham gia vào việc tạo lại mã Hamming. Xác suất n- 1 chia sẻ có thể xây dựng mã Hamming chính xác là 1 / n. Trong trường hợp này, kẻ thù có thể tạo lại mã Hamming với xác suất là 1, do đó, thông tin lớp thứ ba có thể thu được bằng cách . Xác suất để chỉ một cổ phiếu không tham gia vào việc tạo lại mã Hamming là 1- 1/n . Trong trường hợp này, kẻ thù có thể tạo lại mã Hamming với xác suất 0,5, do đó, thông tin lớp thứ ba có thể thu được bằng cách (n - 1) / [ \* ] . Do đó, tổng xác suất giải mã thông tin lớp thứ ba là { \* .

**2.3.4 Thử nghiệm và phân tích**

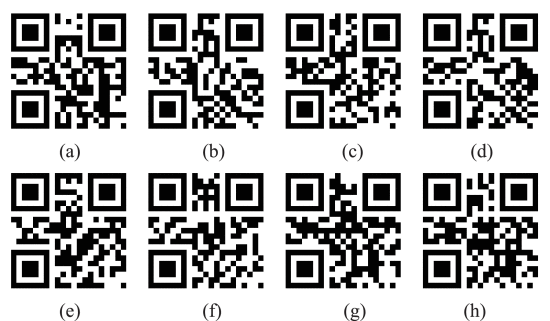
Xác minh thêm về tính khả thi của kế hoạch, bao gồm cả tính mạnh mẽ và trọng tải bí mật, được minh họa trong phần này. Ngoài ra, một số so sánh và phân tích về công việc liên quan được cung cấp.

### **a. Kết quả thí nghiệm**

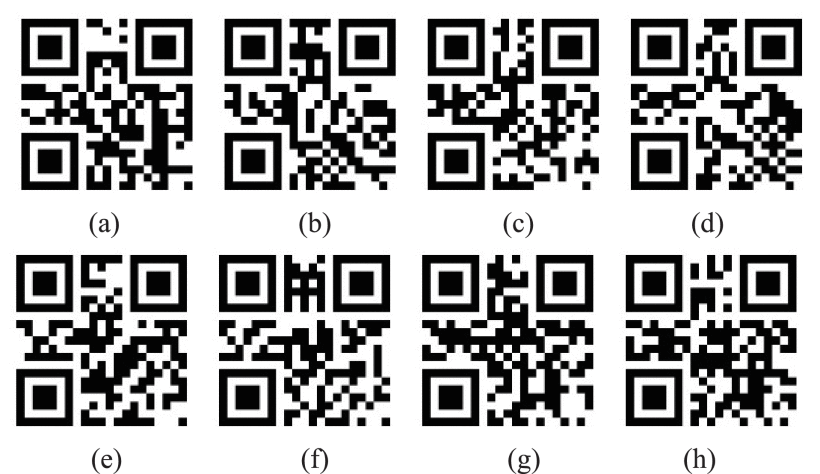
Trong phần này, chúng tôi sử dụng lược đồ (8, 8) làm ví dụ; chúng tôi chọn mã QR của phiên bản 1 với mức sửa lỗi L làm hình ảnh sóng mang, và các luồng 24 bit và 32 bit được đưa vào làm thông tin lớp thứ hai và thông tin lớp thứ ba, tương ứng. Bảng 2 cho thấy tập dữ liệu cho thử nghiệm và các hình ảnh tương ứng của thử nghiệm được đưa ra trong Hình 4 , trong đó (a) - (h) là mã QR của nhà cung cấp dịch vụ với kích thước 250×250 , và (a) - (h) trong Hình 5 là các cổ phiếu có kích thước 250×250.



Bảng 2: Các trường hợp sử dụng thử nghiệm



Hình 4: Mã QR của nhà cung cấp dịch vụ được đề xuất



HÌnh 5: Các cổ phần của chương trình đề xuất.

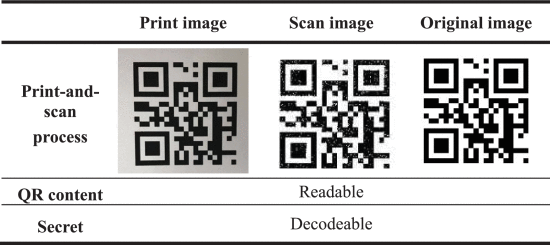
Mã QR của nhà cung cấp dịch vụ được tạo bởi trình tạo mã QR tiêu chuẩn “PsQREdit 2.4.3-Chinese” và các chia sẻ trong Hình 5 có thể được nhận dạng bởi bộ giải mã mã QR tiêu chuẩn hoặc thiết bị có chức năng giải mã; kết quả được công nhận vẫn phù hợp với trường hợp thử nghiệm, điều này chứng tỏ tính hiệu quả của chương trình.

Sơ đồ đề xuất được MATLAB hiện thực hóa, thông qua đó thông tin lớp thứ hai và thông tin lớp thứ ba được trích xuất hoàn toàn, chứng minh tính đúng đắn của sơ đồ.

**b.Kiểm tra và phân tích hiệu suất**

**b.1 Độ bền**

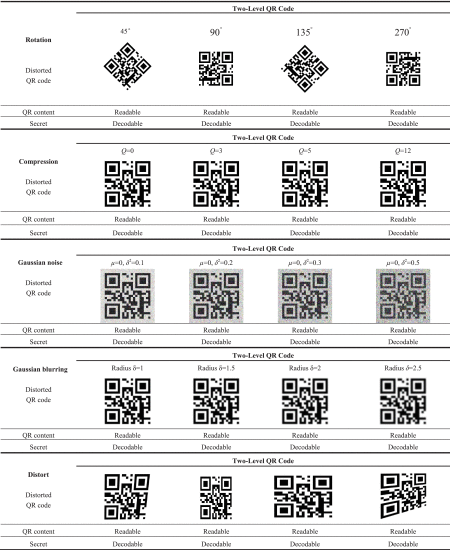
Thao tác in và quét cho phần chia sẻ được thực hành rộng rãi trên các tạp chí thương mại, áp phích và các lĩnh vực xuất bản. Để minh họa rằng lược đồ được đề xuất hỗ trợ các ứng dụng đa mạng, chúng tôi sử dụng máy in HP LaserJet 500 màu M551 để in tại600×600 dpi và quét bằng máy quét HP LaserJet M2727nf tại 200×200 dpi. Kết quả được thể hiện trong Bảng 3 .



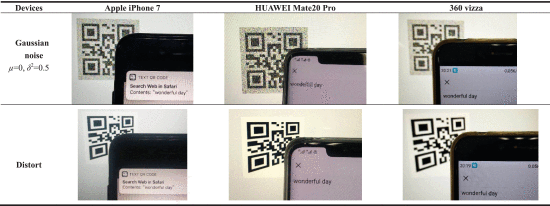
**Bảng 3: Mã QR được đánh dấu sau khi in và quét**

Trong các ứng dụng thực tế, việc bị can thiệp từ bên ngoài là khó tránh khỏi. Để kiểm tra độ chắc chắn của các lượt chia sẻ, chúng tôi phân tích hiệu suất của các lượt chia sẻ được tạo ra khi chúng bị bóp méo bởi nhiều loại nhiễu khác nhau. Xét trường hợp khi thiết bị di động quét lượt chia sẻ, góc quét không cố định, lượt chia sẻ theo các góc khác nhau. Ngoài ra, kỹ thuật nén thường được sử dụng để nén chia sẻ để giảm không gian lưu trữ. Để đánh giá tính khả thi của chương trình được đề xuất, chúng tôi nén phần chia sẻ với các giá trị nén jpg tương ứng là 0, 3, 5 và 12. Hơn nữa, nhiễu luôn xảy ra khi chia sẻ được truyền qua các kênh mở hoặc được thu lại dưới dạng từ thực. Ở đây, MATLAB được sử dụng để thêm nhiễu Gaussian với giá trị trung bình μ = 0 và phương sai . Tính năng làm mờ Gaussian được thêm vào với bán kính 1 pixel, 1,5 pixel, 2 pixel và 2,5 pixel. Ngoài ra, thông tin về cổ phần cũng được quan tâm trong quá trình mua lại. Kết quả sau khi can thiệp ở trên được phân tích và các bài kiểm tra cụ thể được hiển thị trong Bảng 3 , trong đó "Có thể *đọc được* " cho biết rằng chia sẻ đã xử lý có thể được đọc bởi bộ đọc mã QR tiêu chuẩn hoặc các thiết bị có chức năng giải mã và "Có *thể* giải mã " cho biết rằng chia sẻ đã xử lý có thể được sử dụng làm đầu vào cho lược đồ để trích xuất thông tin lớp thứ hai và lớp thứ ba.

Vì số lượng pixel được lật của mỗi mã QR của nhà cung cấp dịch vụ nhỏ hơn C/2 , các chia sẻ được tạo duy trì khả năng đọc của mã QR. Để xác minh khả năng đọc được của tỷ lệ bị bóp méo trong Bảng 4 , chúng tôi sử dụng ba điện thoại di động, Apple iPhone7, HUAWEI Mate20 Pro và 360vizza, để giải mã hai loại tỷ lệ méo mó điển hình trên trong Bảng 4 ; kết quả được thể hiện tron8g Bảng 5 .

****

**BẢNG 4** Kiểm tra độ bền

****

**BẢNG 5** Kết quả giải mã với thiết bị di động

**b.2 Tải trọng bí mật**

Có 2p mã QR của nhà cung cấp dịch vụ có kích thước m \* n ; độ dài của thông tin lớp thứ hai làx bit, độ dài của luồng thông tin lớp thứ ba là Y bit, số từ mã sửa lỗi là LÀ , và tổng số tiền bí mật là *số tiền* . Mô-đun của mã QR thỏa mãn công thức sau [(4)](https://ieeexplore.ieee.org/document/#deqn4-5) .

(4)

**(5)**

Dựa trên sơ đồ được đề xuất, một số pixel có thể bị thay đổi, nguyên nhân là do nhúng luồng thông tin lớp thứ hai. Vị trí của các pixel lật này được sử dụng làm chỉ mục vị trí để nhúng luồng thông tin lớp thứ ba và các pixel lật 1 bit có thể truyềnp -bit thông tin. Tuy nhiên, cơ chế sửa lỗi của mã QR và nội dung của thông tin lớp thứ ba tự nó hạn chế vị trí nhúng. Do đó, thông tin lớp thứ ba phải thỏa mãn công thức [(6)](https://ieeexplore.ieee.org/document/#deqn6) .

(6)

Nếu và chỉ khi, kết quả XOR-ed của mỗi cột của khối pixel sóng mang bằng với luồng thông tin lớp thứ 2 thì Y=0 tồn tại. Ngoài ra, khi tất cả các vị trí của pixel được lật thỏa mãn điều kiện nhúng của luồng thông tin lớp thứ ba,Y= *px* tồn tại.

Công thức [(7)](https://ieeexplore.ieee.org/document/#deqn7-8) và công thức [(8)](https://ieeexplore.ieee.org/document/#deqn7-8) tồn tại khil các bit của kết quả XOR-ed của mỗi cột của khối pixel sóng mang bằng với luồng thông tin lớp thứ hai và nếu có q mã QR của nhà cung cấp dịch vụ được thay đổi sau khi nhúng thông tin lớp thứ ba.

(7)

\* q \* 8 \* (8)

Khi nào l = 0 , công thức [(8)](https://ieeexplore.ieee.org/document/#deqn7-8) chuyển thành công thức [(9)](https://ieeexplore.ieee.org/document/#deqn9)

\* q \* 8 \* (9)

Dựa trên phân tích ở trên, giới hạn trên của tổng khối lượng bí mật của kế hoạch là

\* q \* 8 \* (10)

Trọng tải bí mật tối đa được thực hiện bởi một mã QR của nhà cung cấp dịch vụ duy nhất là payload = ⌊amount/2p⌋ , có thể được mô tả chi tiết như công thức [(11)](https://ieeexplore.ieee.org/document/#deqn11) .

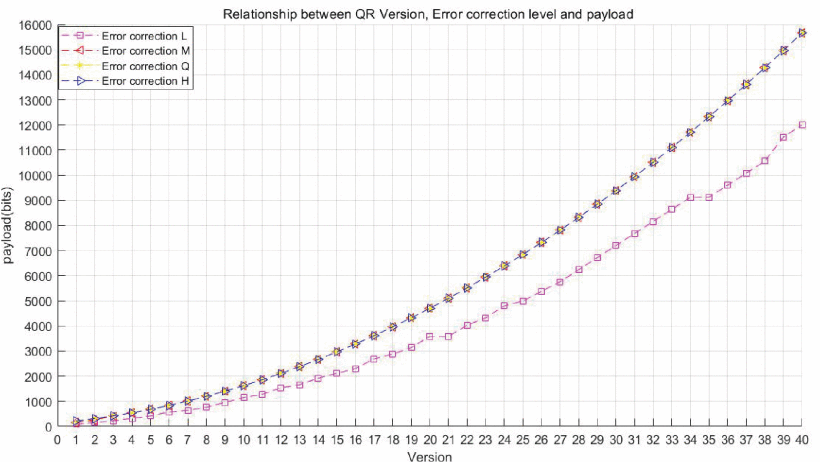
\* q \* 8 \* (11)

Đưa ra một trường hợp mà p = 3 và q = 8 , công thức [(11)](https://ieeexplore.ieee.org/document/#deqn11) có thể được viết dưới dạng công thức [(12)](https://ieeexplore.ieee.org/document/#deqn12) .

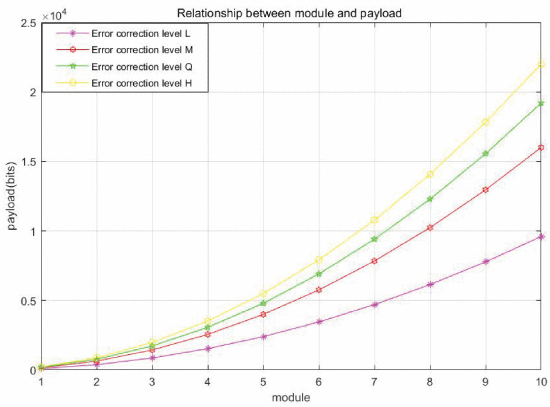
payload = (12)

Từ công thức [(12)](https://ieeexplore.ieee.org/document/#deqn12) , có thể kết luận rằng cả kích thước của mã QR sóng mang và mức độ sửa lỗi đều xác định giới hạn trên của khối lượng thông tin trong sơ đồ đề xuất. Vì kích thước của mã QR được quyết định bởi phiên bản mã QR và kích thước của *mô-đun* mã QR , nên sử dụng phương pháp biến thể kiểm soát để phân tích chi tiết mối quan hệ giữa các yếu tố ảnh hưởng và giới hạn trên của khối lượng thông tin.

Đối với khi kích thước của *mô-đun* được cố định, phần này lấy mã QR của *mô-đun* = 1 làm ví dụ và phân tích mối quan hệ giữa giới hạn trên của tải trọng thông tin và phiên bản của mã QR. Biểu đồ quan hệ được hiển thị trong Hình 6 , trong đó rõ ràng là khi *mô-đun* = 1, giới hạn trên của trọng tải tăng lên khi phiên bản và khả năng sửa lỗi của mã QR tăng lên, với tối thiểu 96 bit.

  
**HÌNH 6:** Mối quan hệ giữa tải trọng và phiên bản.

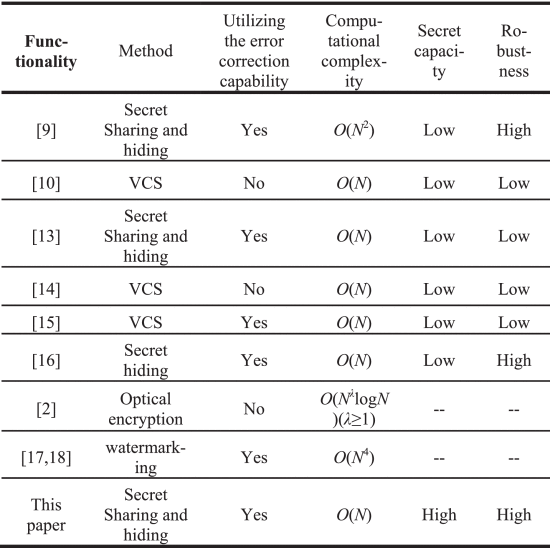
Khi phiên bản được sửa, phần này sử dụng mã QR với phiên bản số 1 làm ví dụ và phân tích mối quan hệ giữa giới hạn trên của trọng tải thông tin, kích thước của *mô-đun* mã QR và mức độ sửa lỗi. Biểu đồ mối quan hệ được thể hiện trong Hình 7 . Có thể kết luận rằng bằng cách điều chỉnh *mô-đun* , giới hạn trên của khối lượng thông tin có thể được cải thiện đáng kể theo phiên bản nhất định và mức sửa lỗi. Khi *mô-đun* được sửa, *tải trọng* tỷ lệ với mức sửa lỗi.



**HÌNH 7:** Mối quan hệ giữa tải trọng, mức sửa lỗi và kích thước mô-đun.

**b.3 So sánh và thảo luận**

So sánh và thảo luận của bài báo này và các chương trình liên quan khác được cung cấp trong phần này. Bảng 6 liệt kê các so sánh tổng thể của bài báo này và các chương trình liên quan khác.



**BẢNG 6** So sánh tổng thể các công trình liên quan

Bảng 6 cung cấp một so sánh tổng thể về các lược đồ ẩn dữ liệu liên quan đến QR, lược đồ đánh dấu thủy tinh, lược đồ mật mã trực quan và các lược đồ chia sẻ bí mật chung, ngoài cách tiếp cận được đề xuất.

##### **a: Phương pháp & Độ phức tạp tính toán của chiết xuất**

Dựa trên công nghệ chia sẻ và che giấu bí mật, Tham khảo [9] , [13] thiết kế mã QR hai cấp mà thông tin công khai có thể được đọc bởi bất kỳ thiết bị nào có chức năng giải mã và thông tin bí mật chỉ có thể được trích xuất bởi cơ quan pháp luật. nhân viên có thẩm quyền. Hoạt động chính của [9] dựa trên hàm băm, có độ phức tạp tính toán làO (N2) . Trong khi [13] chủ yếu bao gồm các phép toán cộng và mô đun, dẫn đến độ phức tạp tính toán thấp nhưO(N) . Dựa trên mật mã trực quan, Tham khảo [10] , [14] , [15] nhận ra việc nhúng thông tin vào mã QR với độ phức tạp tính toán thấp của việc giải mã nhưO(N) . Tham khảo [16] là một kế hoạch che giấu bí mật bằng cách cải thiện khai thác hướng sửa đổi để bảo vệ tin nhắn riêng tư trong mã QR với độ phức tạp tính toán nhưO(N) . Độ phức tạp về thời gian của [2] nơi mã hóa quang học được quyết định bởi tham sốλ . Tham khảo [17] , [18] cả hai đều sử dụng công nghệ khắc chìm kỹ thuật số để thực hiện việc nhúng thông tin bí mật. Bởi vì các lược đồ sử dụng biến đổi cosin rời rạc để thực hiện các phép toán trong miền tần số, độ phức tạp tính toán của nó tương đối cao. Trong sơ đồ được đề xuất, lớp giải mã thông tin thứ hai được hoàn thành bằng một thao tác XOR đơn giản và lớp thông tin thứ ba được trích xuất bằng phép nhân ma trận. Rõ ràng, bài báo của chúng tôi có độ phức tạp về thời gian thấp khi so sánh với một số tác phẩm khác.

##### **b: Sử dụng khả năng sửa lỗi**

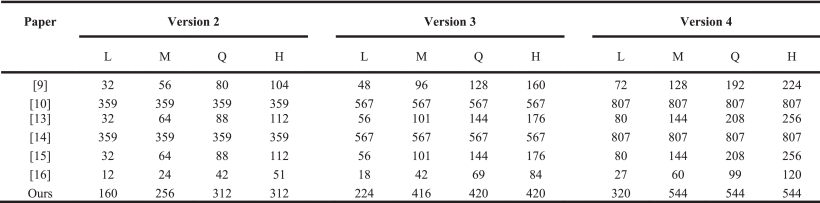
Dựa trên cơ chế sửa lỗi của mã QR, bài báo này và Tham khảo [9] , [13] , [15] , [16] nhận ra việc nhúng thông tin bí mật bằng cách thay đổi các từ mã một phần của mã QR với điều kiện đảm bảo khả năng đọc được mã QR. Tham khảo [10] , [14] thay thế mô-đun của mã QR bằng các dạng khối pixel khác nhau để nhúng thông tin bí mật dựa trên các đặc điểm giải mã của mã QR.

##### **c: Năng lực bí mật**

Để cho p = 3 , và so sánh tải trọng bí mật của sơ đồ được đề xuất với Tham chiếu [9] , [10] , [13] - [14] [15] [16] .

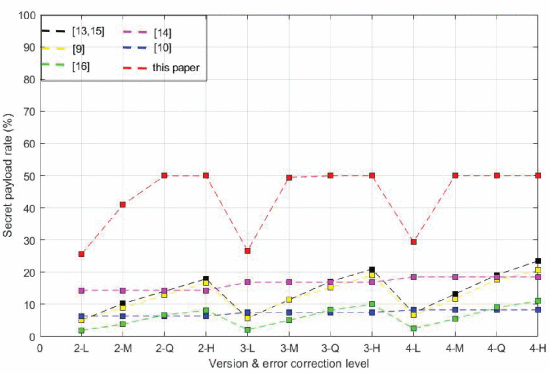
Tham khảo [9] , [13] , [15] , [16] sử dụng khả năng sửa lỗi của mã QR để thực hiện nhúng thông tin bí mật bằng cách sửa đổi một phần pixel của mã QR, điều này giới hạn tải trọng tối đa của thông tin bí mật về khả năng sửa lỗi của mã QR. trong khi bài báo này sử dụng hoạt động XOR và nguyên tắc sửa lỗi mã Hamming để nhận ra việc tái sử dụng các pixel sóng mang. Do đó, trọng tải bí mật của phương án đề xuất cao hơn so với Tham khảo [9] , [13] , [15] , [16] .

Dựa trên đặc tính giải mã của mã QR, Tham khảo [10] , [14] thay thế tất cả các mô-đun không có chức năng của mã QR bằng các mô-đun con của3×3 và 2×2 để lưu trữ thông tin bí mật, tương ứng. Vì vậy, tổng số tiền được nhúng cao hơn chương trình của chúng tôi. Tuy nhiên, lượng chia sẻ được tạo ra lớn hơn 9 lần so với mã QR của nhà cung cấp dịch vụ ở [10] và lớn hơn 4 lần so với mã QR của nhà cung cấp dịch vụ trong [14] . Do đó, lượng thông tin bí mật được mang bởi một pixel đơn vị trong Tham chiếu [10] , [14] thấp hơn của chúng tôi. Kết quả so sánh cụ thể được trình bày trong Bảng 7 .

****

**BẢNG 7** Số lượng tải trọng trong Đề án đề xuất và các công việc liên quan

Để biểu diễn hiệu suất một cách trực quan hơn, chúng tôi xác định tốc độ tải thông tin của hình ảnh sóng mang (tức là số lượng bit thông tin bí mật được mang bởi một pixel đơn vị) để mô tả hiệu suất và đưa ra một biểu đồ như trong Hình 8 , nơi cho thấy bằng chứng rằng tỷ lệ tải trọng bí mật của công việc của chúng tôi cao hơn các công trình khác.

****

**HÌNH 8.** So sánh tỷ lệ tải trọng bí mật.

##### **d: Độ bền**

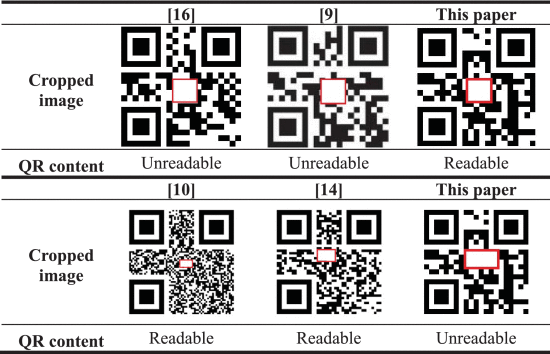
Phần này sẽ so sánh và phân tích hiệu suất chống lại tấn công cắt xén, tấn công nhiễu Gaussian và tấn công nhiễu mờ của phương án đề xuất với các công trình khác.

Đề án được đề xuất và Tham khảo [9] , [13] , [15] , [16] hoạt động trực tiếp trên mô-đun của mã QR, và sử dụng khả năng sửa lỗi của chính mã QR để nhúng thông tin bí mật. Sự khác biệt là Tham chiếu [9] , [16] sử dụng tất cả khả năng sửa lỗi của một mã QR của nhà cung cấp dịch vụ duy nhất để đạt được khả năng nhúng tối đa thông tin bí mật. Nhưng trong nghiên cứu của chúng tôi và Tài liệu tham khảo [13] , [15] , khi thông tin bí mật được nhúng, nó sẽ chiếm một phần khả năng sửa lỗi của một bộ mã QR của nhà cung cấp dịch vụ. Do đó, bài báo này và [13] , [15]có độ bền cao hơn Ref [9] , [16] chống lại sự tấn công cắt xén.

Tham khảo [10] , [14] nhúng bí mật bằng cách thiết kế các đơn vị nhận dạng khác nhau của mã QR mà không chiếm khả năng sửa lỗi của mã QR, dẫn đến khả năng chống lại cuộc tấn công cắt xén mạnh hơn so với Tham khảo [9] , [13] , [ 15] , [16] và phương án đề xuất. Tuy nhiên, trong trường hợp có cùng kích thước, các đơn vị nhận dạng có kích thước3×3 trong [10] , và các đơn vị nhận dạng có kích thước2×2 trong [14] làm cho các bài chia sẻ có mật độ pixel cao hơn so với bài báo này và Tham khảo [9] , [13] , [15] , [16] , dẫn đến khả năng chống lại các cuộc tấn công nhiễu Gaussian và nhiễu mờ yếu hơn bài báo này và Tham khảo [9] , [13] , [15] , [16] .

Để kiểm tra độ chắc chắn chống lại cuộc tấn công cắt xén, chúng tôi so sánh sơ đồ được đề xuất với các Tham chiếu [9] , [16] và Tham chiếu [10] , [14] tương ứng. Cả hai thử nghiệm đều chọn tỷ lệ phiên bản 1 và kích thước250×250 với mức sửa lỗi L làm đối tượng thí nghiệm.

Đối với phần được tạo bởi bài báo này và Tham chiếu [9] , [16] , hãy cắt khối pixel với kích thước50×50 ở cùng một vị trí. Và cắt khối pixel trong bài báo này và Tham chiếu [10] , [14] với kích thước70×40 ở cùng một vị trí. Điều đáng nói là quy mô của phần trong bài báo này và Tham khảo [9] , [16] là250×250 và 750×750 trong [10] ,500×500 trong [14] . Kết quả thí nghiệm được cung cấp trong Bảng 8 .

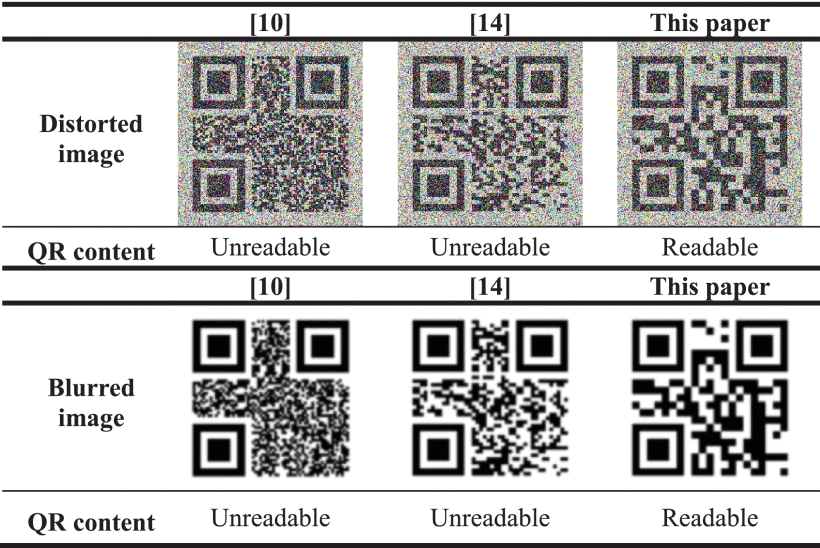


**BẢNG 8** So sánh độ bền chống lại cuộc tấn công cắt xén

Kết quả thực nghiệm cho thấy rằng tính mạnh mẽ chống lại sự tấn công cắt xén của sơ đồ đề xuất yếu hơn so với Tham chiếu [10] , [14] và mạnh hơn so với Tham chiếu [9] , [16].

Khi hình ảnh bí mật được khôi phục, một phần thông tin bí mật của Tham chiếu [9] , [10] , [14] , [16] và một phần thông tin lớp thứ hai của sơ đồ được đề xuất không thể được giải mã chính xác. Trong khi khi phần chia sẻ bị hỏng không tham gia vào quá trình nhúng thông tin lớp thứ ba, thì việc trích xuất thông tin lớp thứ ba của lược đồ không bị ảnh hưởng.

Để kiểm tra độ chắc chắn chống lại cuộc tấn công nhiễu của sơ đồ được đề xuất, phần này bổ sung tiếng ồn trắng Gauss với giá trị trung bình μ = 0 và phương sai của d2=0.5 vào phần được tạo bởi Ref [10] , [14] và làm mờ phần được tạo bởi Gaussian Blur với bán kính 1,5. Kết quả thí nghiệm được trình bày trong Bảng 9 .



**BẢNG 9** Phân tích so sánh về độ bền chống lại cuộc tấn công bằng tiếng ồn

Các kết quả thực nghiệm cho thấy rằng khả năng chống lại tấn công nhiễu Gaussian và tấn công nhiễu mờ của phần được tạo ra bởi sơ đồ đề xuất là mạnh hơn so với tham chiếu [10] , [14] .

Dựa trên kết quả thực nghiệm và so sánh với các công trình liên quan, các ưu điểm chính của đề án được đề xuất được tóm tắt như sau:

*Lưu trữ thông tin ba lớp:* Đề án được đề xuất có thể đạt được mục tiêu nhúng thông tin ba lớp bằng thuật toán chia sẻ và thuật toán giải mã mã Hamming.

*Tải trọng bí mật lớn:* Đề án nhận ra việc sử dụng lại các pixel sóng mang bằng cách sử dụng cơ chế sửa lỗi của mã Hamming, để sửa đổi 1 bit có thể truyền bổ sungp -bit thông tin và tải trọng bí mật được cải thiện bằng cách nâng cao hiệu quả nhúng. Qua so sánh và phân tích thực nghiệm, tỷ lệ trọng tải thông tin bí mật tối đa của phương án đề xuất cao hơn [10] , [13] - [14] [15] [16] .

*Độ phức tạp tính toán thấp:* Đề án chủ yếu dựa trên phép toán “XOR” và phép toán nhân ma trận để thực hiện việc khai thác thông tin bí mật. So với sơ đồ giấu thông tin truyền thống và sơ đồ thủy vân, sơ đồ này có ưu điểm là tính toán giải mã có độ phức tạp thấp.

**2.3.5 Phần kết luận**

Bài báo này đề xuất một mã QR ba lớp mới lạ, phong phú với độ phức tạp tính toán thấp. Kế hoạch này làm giảm khả năng bị nghi ngờ bởi những kẻ tấn công tiềm năng khi cổ phiếu được phân phối qua các kênh công khai. Khác với các kế hoạch khác, cách tiếp cận được đề xuất nhúng bí mật dựa trên mô-đun của mã QR. Kết quả thử nghiệm cho thấy sơ đồ này chống lại các cuộc tấn công xử lý hậu kỳ hình ảnh thông thường hơn và nó có thể chống lại hoạt động in và quét một cách đặc biệt. Xem xét các ứng dụng thực tế, chương trình của chúng tôi có thể được sử dụng để đạt được việc quản lý phân loại thông tin. Ví dụ, để bảo vệ hồ sơ bệnh án của bệnh nhân, bệnh viện có thể phân biệt bác sĩ và các nhân viên khác với các giới hạn quyền hạn khác nhau để thông tin của bệnh nhân không thể bị tiết lộ một cách ác ý. Cách tiếp cận này cũng có thể được sử dụng để bảo vệ thông tin riêng tư trong dịch vụ hậu cần nhanh. Ngoài ra, theo đặc điểm của việc lưu trữ thông tin ba lớp trong lược đồ của chúng tôi, thông tin lớp thứ hai có thể được sử dụng để xác thực danh tính và thông tin lớp thứ ba được sử dụng để truyền thông tin nhạy cảm nhằm cải thiện hiệu suất của lược đồ. Tuy nhiên, cấu trúc truy cập bị hạn chế. Làm thế nào để sử dụng công nghệ giấy ướt để đạt được mục đích khai thác mù lớp thông tin thứ ba và tối ưu hóa cấu trúc truy cập vào cấu trúc truy cập chung vẫn là những vấn đề cần giải quyết. và thông tin lớp thứ ba được sử dụng để truyền thông tin nhạy cảm nhằm cải thiện hiệu suất của chương trình. Tuy nhiên, cấu trúc truy cập bị hạn chế. Làm thế nào để sử dụng công nghệ giấy ướt để đạt được mục đích khai thác mù lớp thông tin thứ ba và tối ưu hóa cấu trúc truy cập vào cấu trúc truy cập chung vẫn là những vấn đề cần giải quyết. và thông tin lớp thứ ba được sử dụng để truyền thông tin nhạy cảm nhằm cải thiện hiệu suất của chương trình. Tuy nhiên, cấu trúc truy cập bị hạn chế. Làm thế nào để sử dụng công nghệ giấy ướt để đạt được mục đích khai thác mù lớp thông tin thứ ba và tối ưu hóa cấu trúc truy cập vào cấu trúc truy cập chung vẫn là những vấn đề cần giải quyết.

# **CHƯƠNG 3: NỀN TẢNG REACT NATIVE VÀ LẬP TRÌNH THỬ NGHIỆM**

# **3.1 Tổng quan về lập trình ứng dụng di động React Native**

**3.1.1 Native và Hybird là gì?**

Native App là các ứng dụng được phát triển và build trên chính những công cụ hỗ trợ mà nhà phát triển iOS và Android cung cấp cho người lập trình viên ở đây là XCode và Android. Chúng được viết bằng ngôn ngữ mà nhà phát triển iOS và Android cung cấp ra iOS thì là Swift hiện tại, xưa là Objective C, còn Android thì hiện tại tồn tại song song là Java và Kotlin. Các ứng dụng Native được build dựa trên ngôn ngữ của chính hệ điều hành và sử dụng những tính năng sẵn có của hệ điều hành mà không phải thông qua bất cứ ứng dụng bên thứ ba hay engine nào khác để vận hành nên tốc độ là nhanh nhất.

Hybrid App là sự kết hợp giữa ứng dụng Web và ứng dụng mobile. Tức là chúng vừa có thể cài đặt lên điện thoại người dùng giống như là những ứng dụng Native bình thường và chúng ta có thể tìm thấy chúng trên chợ ứng dụng nhưng khó có thể nhận ra đâu là ứng dụng Native và đâu là ứng dụng Hybrid thường thì người dùng sẽ không để ý quá nhiều. Nhược điểm của Hybrid App đó chính là vấn đề hiệu năng sẽ bị ảnh hưởng đáng kể cũng như không tương tác được hết những tài nguyên hệ thống phần mềm cũng như phần cứng.

**3.1.2. Tại sao phải là React Native?**

React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động. Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai hệ sinh thái luôn. Một điểm hay ho nữa mà em có đề cập là giảm chi phí recompile của Native bằng cách sử dụng Hot-Loading tức là mình không cần phải build lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ quá nhiều thời gian trong việc build và run ứng dụng nữa.  
 Và điểm lợi hại kế tiếp của React Native đó chính là chúng ta chỉ cần sử dụng JS để phát triển được một ứng dụng di động hoàn chỉnh, đồng thời giải quyết được các vấn đề mà Native App gặp phải mà mình đã nêu ở trên. Và rồi còn cả kết hợp với code native như Swift, Java, v.v…

React Native mở ra cơ hội cơ hội việc làm rất nhiều cùng với cộng đồng hỗ trợ rất mạnh mẽ và đông đảo áp đảo so với những công cụ lập trình di động đa nền tảng khác hiện nay. Dẫn chứng là rất nhiều app nổi tiếng hiện nay đều được xây dựng dựa trên React Native. Ví dự như: Facebook, Instagram, Airbnb, Skype, Tesla, Walmart, Discord…Tại Việt Nam ví dụ như: Shopee, Grab, Chợ tốt, Tiki, Lazada…

## 3.1.3 Ưu điểm và nhược điểm

***Ưu điểm:***

* Hiệu quả về mặt thời gian khi muốn phát triển một ứng dụng nhanh chóng.
* Hiệu năng tương đối ổn định.
* Cộng đồng phát triển mạnh.
* Tiết kiệm chi phí.
* Ứng dụng tin cậy và ổn định
* Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất.
* Trải nghiệm người dùng tốt hơn là hybrid app.

***Nhược điểm:***

* Vẫn đòi hỏi native code.
* Hiệu năng sẽ thấp hơn với app thuần native code.
* Bảo mật không cao do dựa trên JS.
* Quản lý bộ nhớ.
* Khả năng tùy biến cũng không thực sự tốt đối với một vài module.

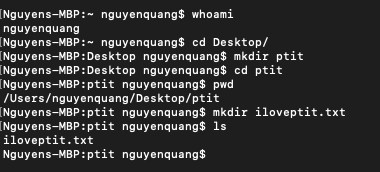
**3.1.4 Công cụ hỗ trợ lập trình React Native**

React Native có thể chạy trên Windows, Linux, MacOS:

+) Để chạy ứng dụng trên hệ điều hành MacOS ta cần phải cài đặt Node.js, Watchman, Terminal, và Xcode cho IOS. Cài đặt Android Studio, Genymotion cho hệ điều hành Android. Sau đó cài đặt môi trường React Native bằng các câu lệnh trong Terminal.

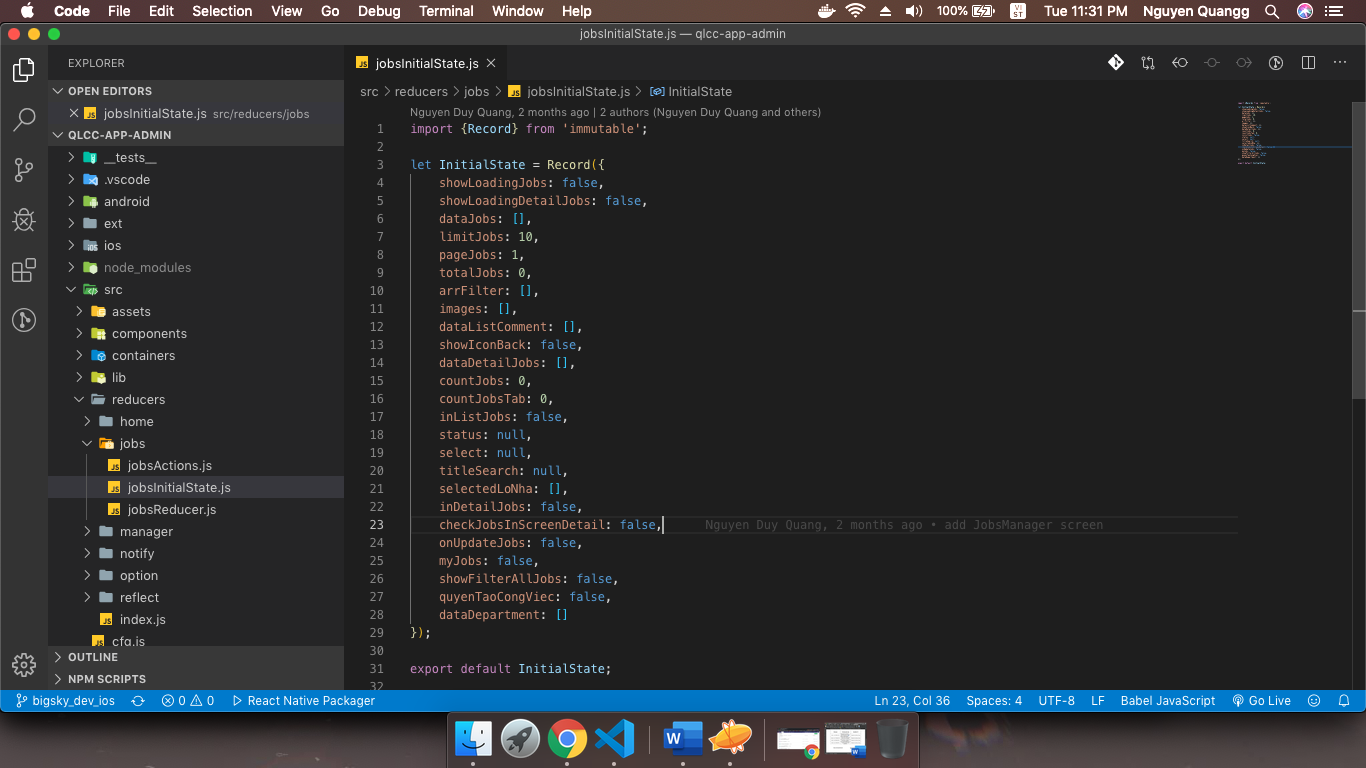
+) Để chạy ứng dụng trên hệ điều hành Linux như Ubuntu và Windowns ta cần phải cài đặt Android Studio, Genymotion, Node.js. Sau đó cài đặt môi trường React Native bằng các câu lệnh trong Terminal đối với Linux hoặc Command Line đối với Windows. Không thể mở máy ảo IOS do Facebook chưa hỗ trợ.

Terminal là giao diện dòng lệnh (Command line interface) và cho phép điều khiển máy tính của mình bằng dấu nhắc lệnh (command prompt). Trong công việc thường ngày dùng để mở file bằng câu lệnh, làm việc với git, push code lên các hệ điều hành IOS, Android…



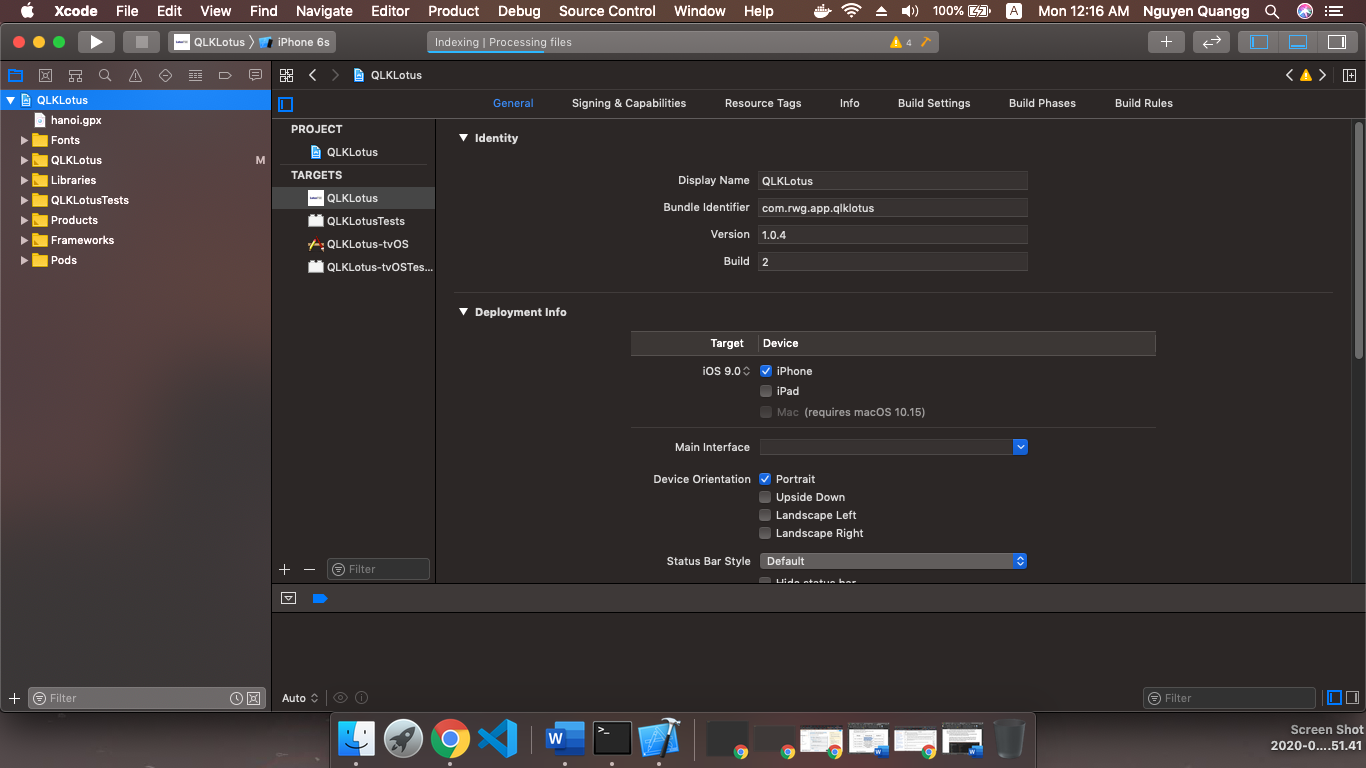
*Một số lệnh cơ bản bằng Linux trong Terminal*

**Visual Studio Code** (VS Code hay VSC) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi.



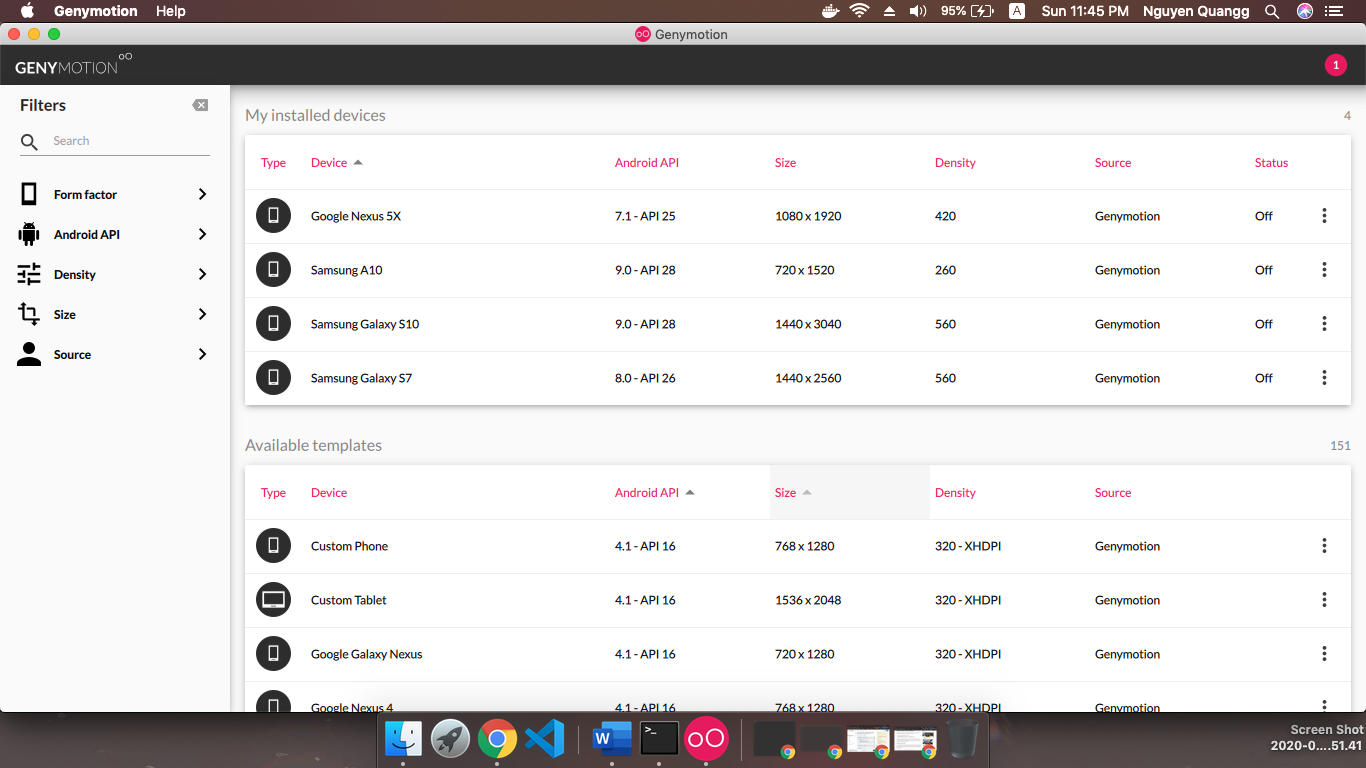
***Visual Studio Code****hỗ trợ nhiều plugin và dễ sử dụng*

Xcode là phần mềm hỗ trợ lập trình cũng như chạy máy ảo ở trên hệ điều hành IOS.

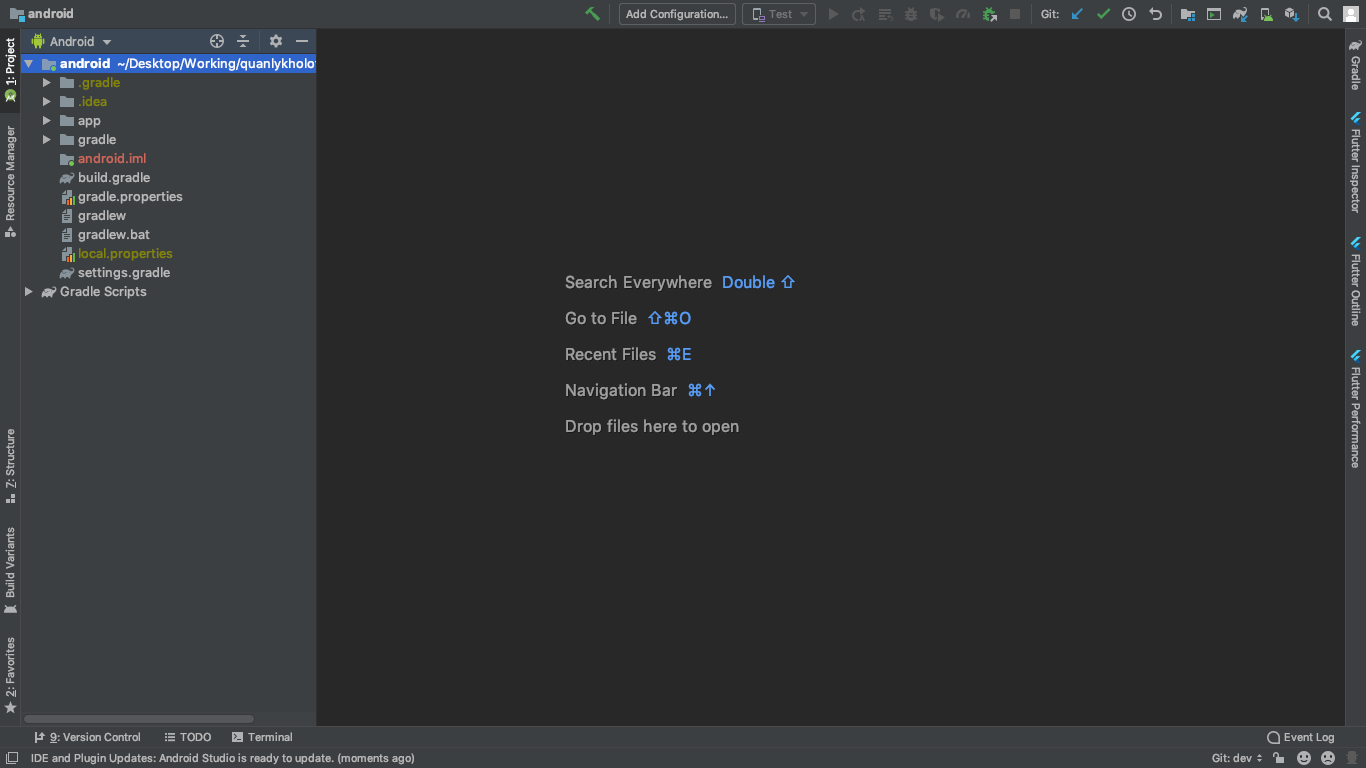


*Giao diện chính của Xcode*

Genymotion và Android Studio là phần mềm hỗ trợ lập trình cũng như chạy máy ảo ở trên hệ điều hành Android.

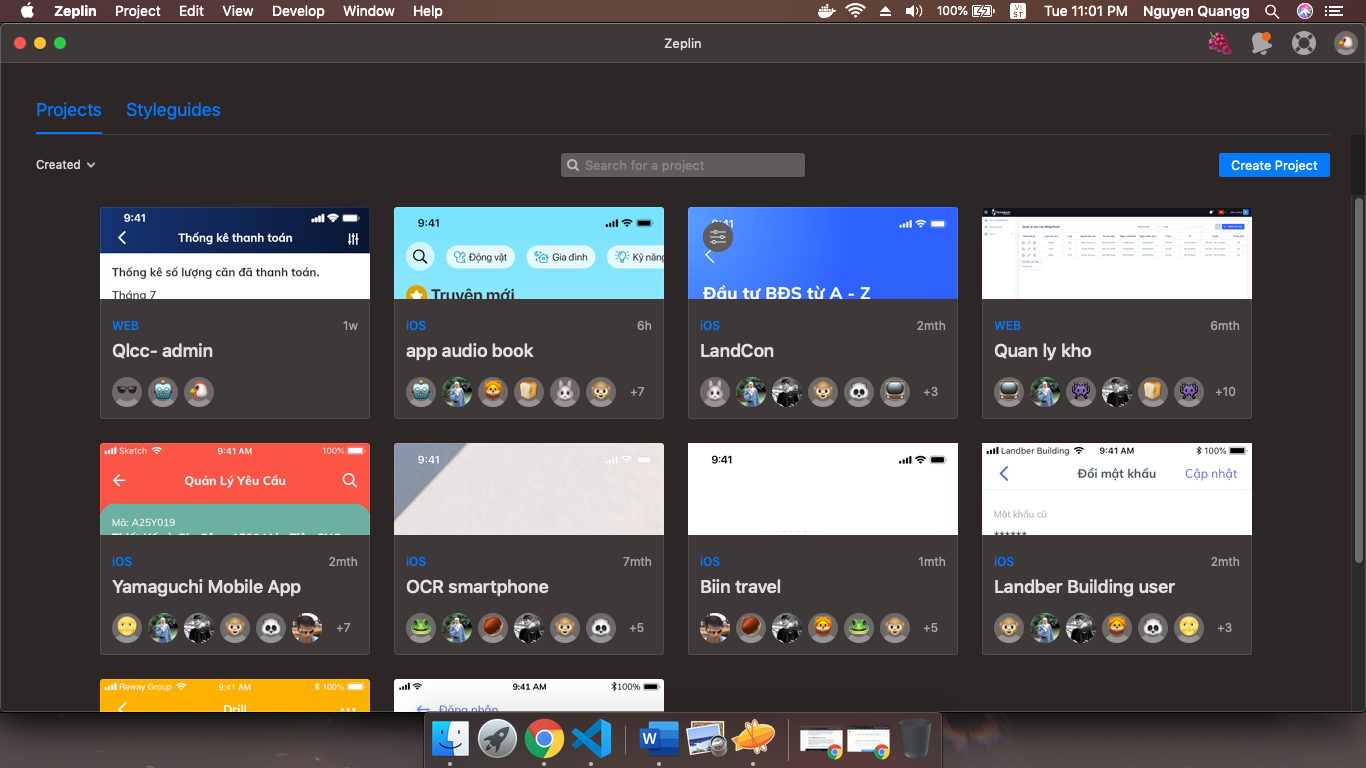


*Giao diện chính của Genymotion*



*Giao diện chính của Android Studio*

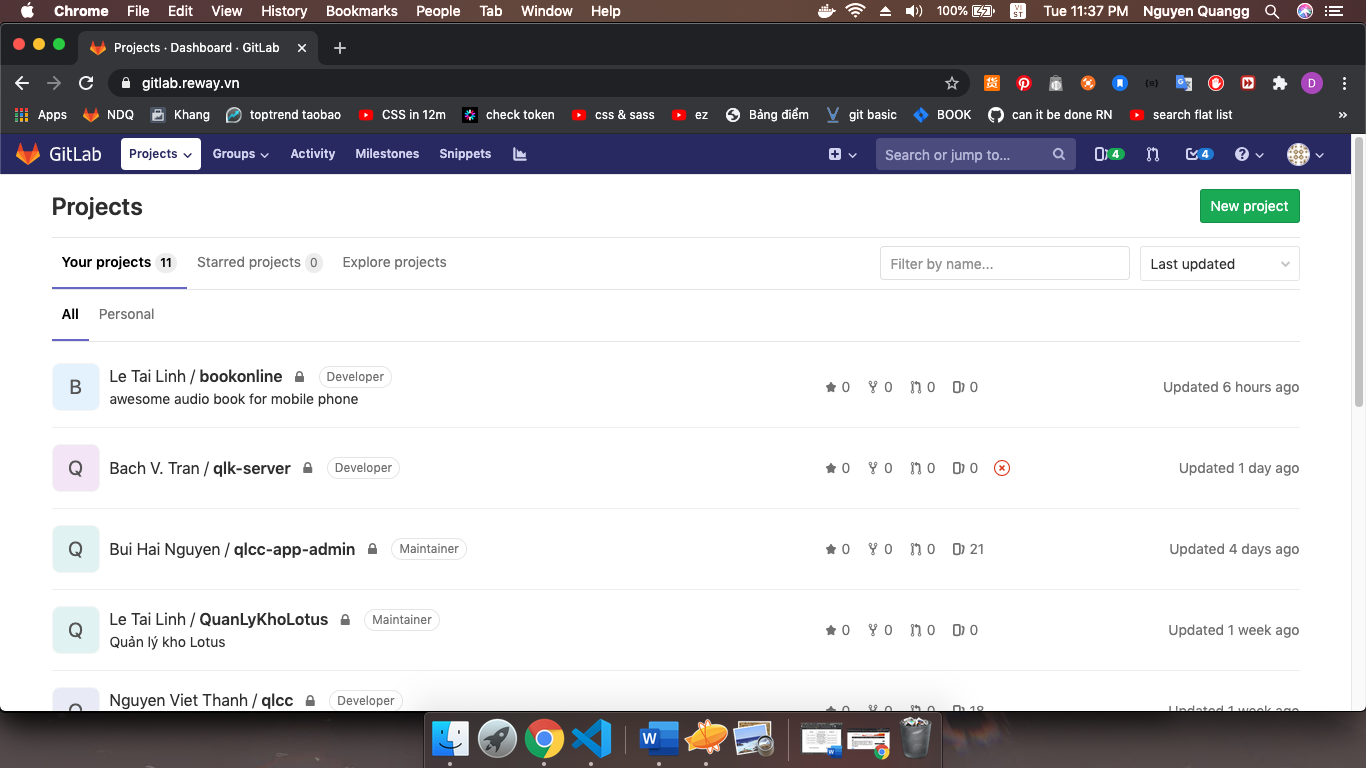
Zeplin là công cụ để kết nối giữa Frontend, Backend và Designer. Zeplin là thứ mà có thể kết nối tất cả hệ điều hành: Window, Ubuntu, Mac. Đối với Developer như lập trình ứng dụng React Native thì chỉ cần cài đặt, được mời vào dự án và có thể xem kích thước ảnh, chữ, màu sắc, căn lề… để lập trình.



*Giao diện chính của Zeplin*

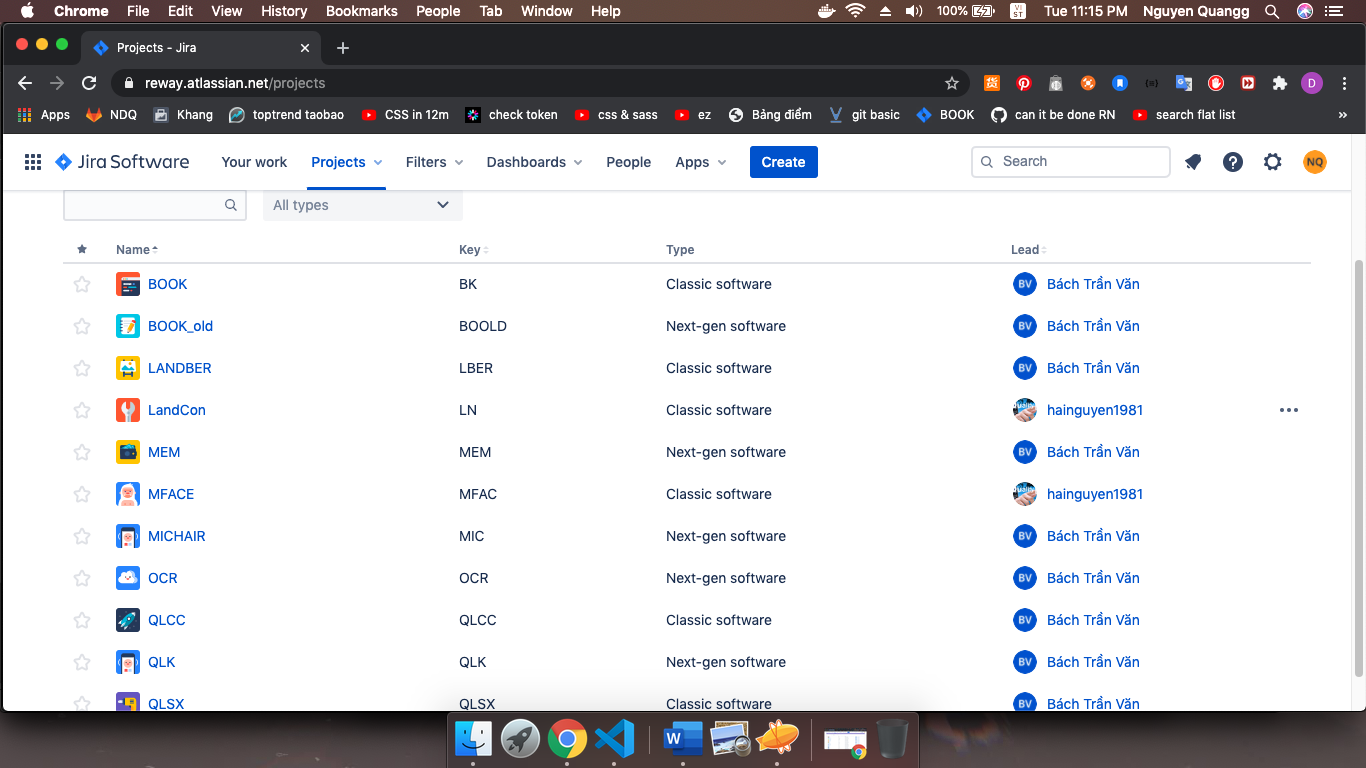
Gitlab là hệ thống self-hosted mã nguồn mở dựa trên hệ thống máy chủ Git dùng để quản lý mã nguồn của bạn. Chức năng chính của Gitlab là chia nhánh và đẩy code lên các nhánh được yêu cầu của dự án để giao tiếp giữa các lập trình viên. Hiện nay có 3 phiên bản chính của Gitlab là:

* ***Gitlab community edition****(****CE****)* là phiên bản cộng đồng, mã nguồn mở, cung cấp qua Git từ kho lưu trữ chứa GitLab. Bản mới nhất của GitLab được các nhà phát triển release tại các nhánh stable và nhánh Master.
* ***GitLab enterprise edition****(****EE****)* là Gitlab phiên bản doanh nghiệp, có sẵn không lâu sau khi phát hành bản CE, được cung cấp từ kho lưu trữ của gitlab.com. Khi một doanh nghiệp đăng ký GitLab sẽ nhận được sự support của GitLab BV khi gặp khó khăn trong quá trình cài đặt và sử dụng
* ***Gitlab continuous integration****(****CI****)* là một giải pháp tích hợp được thực hiện bởi nhóm phát triển Gitlab.

****

*Gitlab phiên bản cho doanh nghiệp*

Jira là một ứng dụng theo dõi và quản lý lỗi / vấn đề trong dự án, được phát triển bởi công ty phần mềm Atlassian của Australia. Jira có chức năng chính như phân quyền cực kỳ chi tiết, không chỉ phân quyền trong dự án chung, mà còn phân quyền đối với từng nhiệm vụ, giúp team công nghệ bảo vệ thông tin độc quyền của mình và là cách giao tiếp chính giữa các Deverloper, Tester, PM…

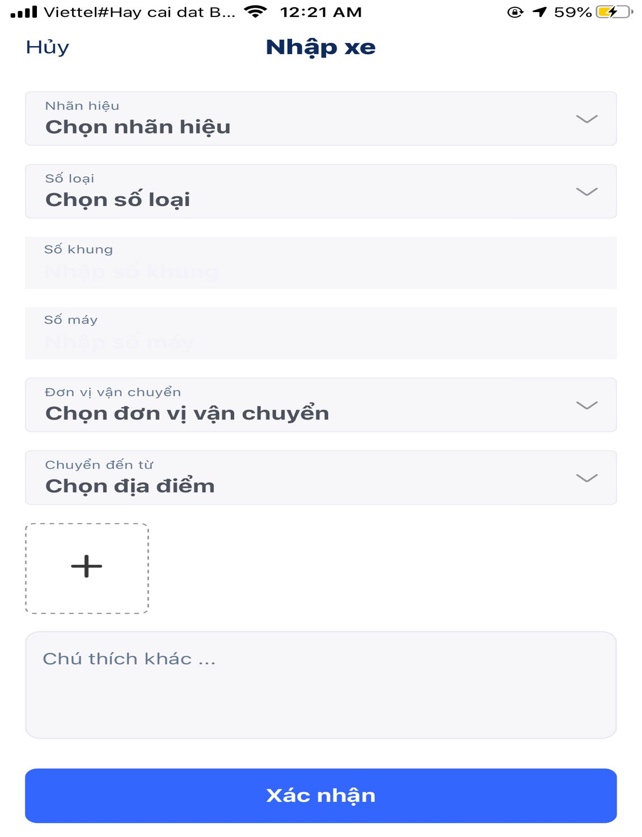


*Các dự án đã và đang làm trên Jira*

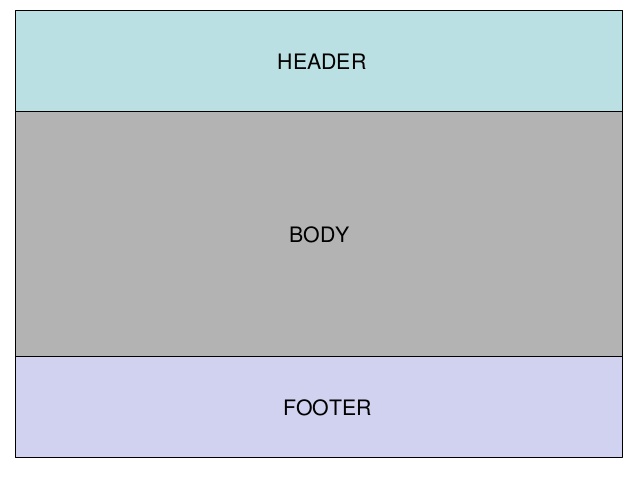
# **3.2 Thành phần chính của React Native**

## 3.2.1 Components

Components là một khái niệm cơ bản của cả React và React native. Chính việc chia nhỏ ứng dụng thành các components nhỏ tạo nên tính tái sử dụng cao và khả năng mở rộng của chúng. Hãy thử phân tích một ví dụ sau đây:



Đây là một màn hình nhập xe vào kho gồm có các label như: nhãn hiệu, số loại, số khung, số máy, đơn vị vận chuyển, chuyển đến từ. Ngoài ra còn có thể chụp ảnh từ Camera hoặc chọn ảnh trong anbuml, chú thích nếu có và nút xác nhận nhập xe vào kho. Chúng ta có thể hình dung ra rằng cả màn đó sẽ là một Scroll View và các item trong đó là biểu thị cho thành phần của xe được nhập vào kho. Đây có thể là một cách chia các component cho ví dụ này.



Ngoài những component chúng ta xây dựng và sử dụng lại đó như Header và Footer, thì React Native sẵn có đó là một tá các component mặc định ở <https://reactnative.dev/docs/components-and-apis>.

## 3.2.2 Props

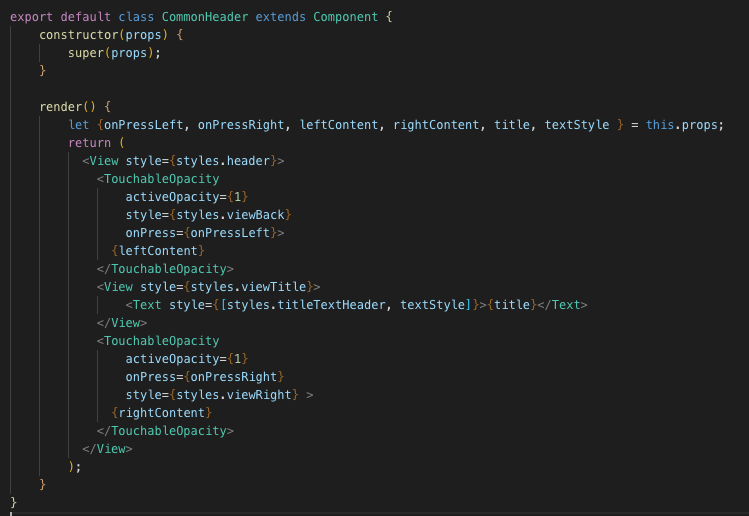
Props là viết tắt của Properties. Một điều mà mình cần phải nhớ khi sử dụng props đó là không bao giờ nên thay đổi giá trị của nó, hay nói cách khác, đây là một dữ liệu immutable (bất biến).

Các component nhận props từ component cha. M không được thay đổi giá trị của props trong các component này mà chỉ được phép đọc giá trị ra thôi. Trong React thì dữ liệu sẽ đi theo một chiều, có nghĩa là từ component cha => các component con.

Đây là một ví dụ em sử dụng props, cung cấp cho một component tên là CommonHeader:



Trong component CommonHeader em sẽ sử dụng props title này để render ra tiêu đề của nó:

****

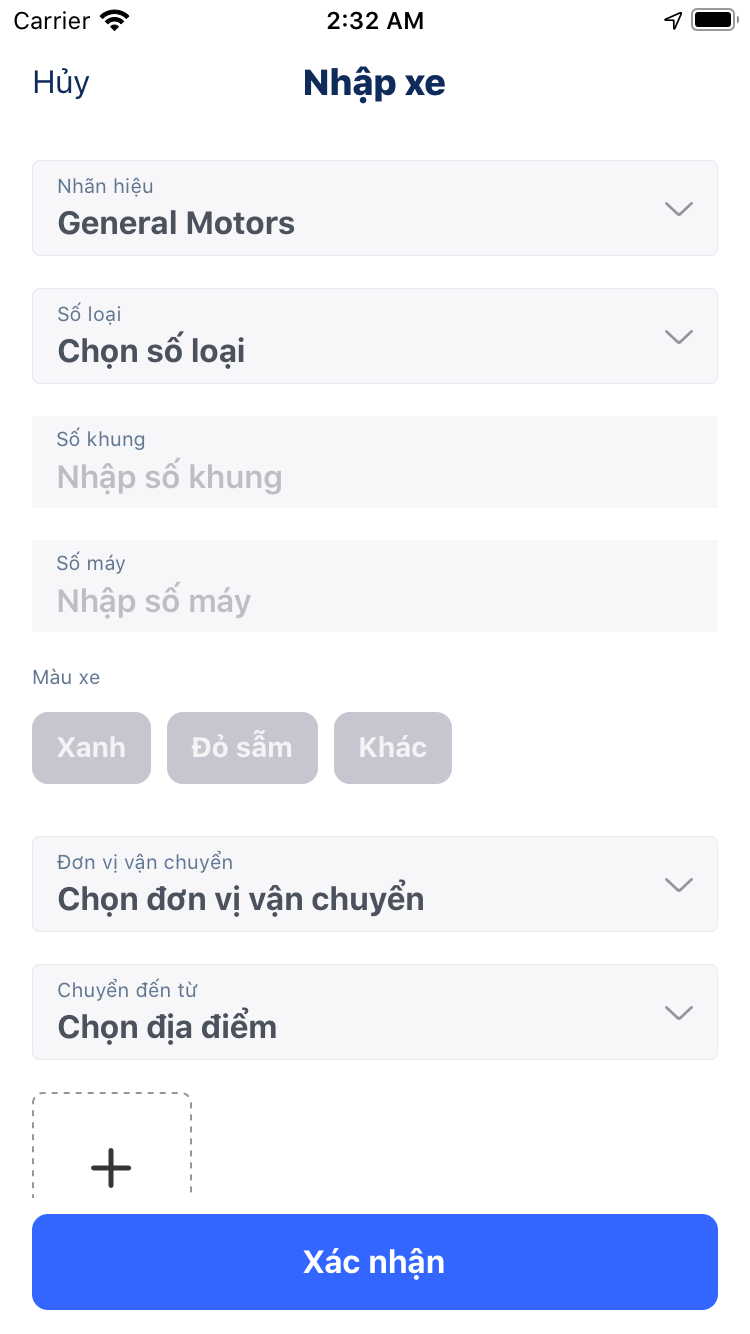
## 3.2.3 State

State thì hoạt động khác với Props. State là dữ liệu nội bộ của một Component, trong khi props là dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state, và coi nó là một kiểu dữ liệu mutable. Vì đặc điểm này nên chúng ta hay sử dụng State để thay đổi dữ liệu của view, binding data lại view khi có thay đổi. Nhưng chúng ta không dùng this.state để gán lại giá trị thay đổi cho nó, mà chúng ta sẽ dùng this.setState. Function này sẽ trigger cho class rằng hãy render lại component và các component con của nó, còn this.state thì không.

Đây là một ví dụ về sử dụng state để thay đổi giá trị “itemBrand” từ null thành có giá trị để hiển thị danh sách màu xe cho user chọn màu xe.



*State để thay đổi giá trị itemBrand*



*itemBrand có giá trị là: General Motors*

# **3.3 Kết nối API,** **Redux và Firebase vào dự án thực tế**

**3.3.1 API là gì?**

API là viết tắt của Application Programming Interface – phương thức trung gian kết nối các ứng dụng và thư viện khác nhau. Nó cung cấp khả năng truy xuất đến một tập các hàm hay dùng, từ đó có thể trao đổi dữ liệu giữa các ứng dụng. Thi thoảng vẫn có người lầm tưởng API là một ngôn ngữ lập trình nhưng thực ra API chỉ là các hàm hay thủ tục thông thường. Các hàm này được viết trên nhiều ngôn ngữ lập trình khác nhau.

## 3.3.2 Ưu và nhược điểm của API

***Ưu điểm:***

* Giao tiếp hai chiều phải được xác nhận trong các giao dịch sử dụng API. Cũng chính vì vậy mà các thông tin rất đáng tin cậy.
* API là công cụ mã nguồn mở, có thể kết nối mọi lúc nhờ vào Internet.
* Hỗ trợ chứng năng RESTful một cách đầy đủ.
* Cung cấp trải nghiệm thân thiện với người dùng

***Nhược điểm:***

* Tốn nhiều chi phí phát triển, vận hành, chỉnh sửa.
* Đòi hỏi kiến thức chuyên sâu.
* Có thể gặp vấn đề bảo mật khi bị tấn công hệ thống.

## 3.2.3 Cách dùng API trong dự án thực tế

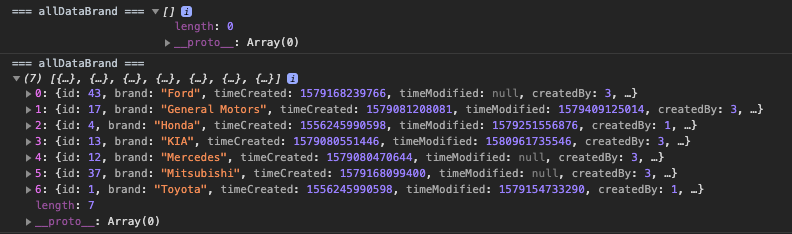
Giả sử ta cần lấy danh sách các loại xe của một kho ô tô, ta cần thiết lập cách gọi API đến máy chủ bằng hàm *getDanhMucBrand* trong thư mục *userApi* như sau*:*

****

Để đọc được dữ liệu hàm *getDanhMucBrand* hay bất kỳ các hàm khác trong thư mục *userApi* ta cần phải dùng state. Ban đầu ta khai báo state là allDataBrand rồi cho nó giá trị bằng mảng rỗng và sau đó được trả về dữ liệu sau khi gọi API. Tất cả các API muốn dùng được đều phải gọi trong hàm componentDidMount như hình dưới đây:



Còn đây là hình ảnh consoleLog ở trên debuger, chúng ta có thể thấy rõ ràng lúc đầu state.allDataBrand vẫn còn rỗng và sau đó khi gọi API, và có data trả về, nhờ vào việc setState lại view có thể render và hiển thị ra ngoài.



# **3.4 Kết nối Redux vào dự án Lotus – Quản lý kho**

## 3.4.1 Redux là gì và các thành phần của Redux

**Redux** là một *predictable state managent tool* cho các ứng dụng Javascript. Nó giúp ta viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. Redux ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ **Elm** và kiến trúc **Flux**của Facebook. Do vậy Redux thường dùng kết hợp với React và React Native.

Có 3 thành phần của Redux: Actions, Store, Reducers.

* Actions đơn giản là các events. Chúng là cách mà chúng ta send data từ app đến Redux store. Những data này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.
* Store là nơi lưu các state của ứng dụng và ta có thể truy cập đến state được lưu, update state thông qua dispatch các actions.
* Reducers là các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như những objects và chúng định rõ cách state của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store.

## 3.4.2 Cài đặt, cấu hình và cách dùng Redux

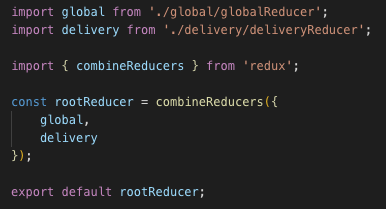
Để dùng được Redux trong dự án này ta cần cài đặt các thư viện chính gồm có redux, redux-thunk, react-redux vào trong dự án.

React-redux dùng để kết nối giữa store và các màn hình với nhau. Redux-thunk là một middleware phổ biến nhất được dùng để xử lý các action bất đồng bộ trong Redux.

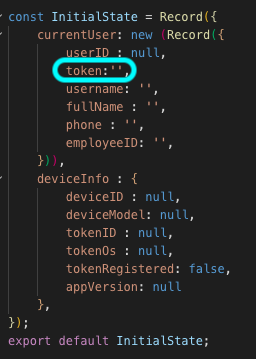
Dưới đây là cấu hình chung cho store cho tất cả các dự án nào mà sử dụng Redux.



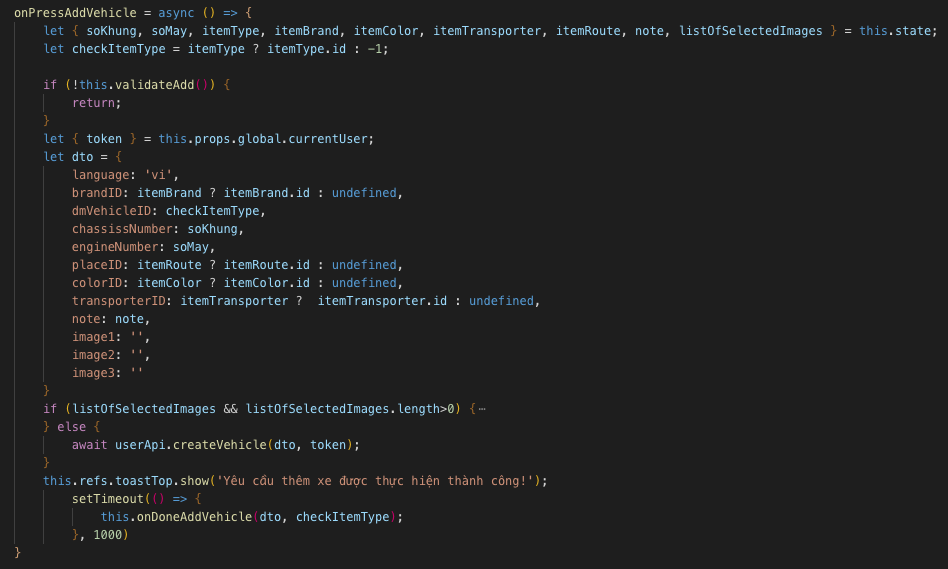
Muốn đẩy lên store ta cần phải có 1 file tổng hợp các file nhỏ như ví dụ dưới đây là 2 reducer có tên là global và delivery. Trong mỗi file đều có 3 file state riêng, actions riêng, reducer riêng và tất cả đều chia nhỏ để dễ quản lý, bảo trì và nâng cấp dự án dễ dàng hơn nếu cần thay đổi.



Như màn nhập xe ở các ví dụ trước, trước khi ta muốn bấm nút xác nhận các thông tin trên đã chọn và nhập xong của xe rồi cần phải có token của người đang thực hiện các thao tác đó. Vậy nên ta cần phải lưu token vào store sau khi người dùng đã đăng nhập vào ứng dụng và mỗi người chỉ có 1 token duy nhất. Tất cả các state của global được khai báo trong file globalInitialState như dưới đây:



Sau khi đăng nhập thành công vào ứng dụng quản lý kho và state token đã được lưu vào store rồi. Nếu muốn lấy token của người đang sử dụng ta chỉ cần gọi lên store bằng câu lệnh this.props.global.currentUser.token. Vậy để xác nhận nhập xe thành công sau khi người dùng đã thao tác cần phải có các giá trị người dùng vừa chọn, token của người dùng và gọi API để cập nhật lên server thành công như hình dưới đây:



# **3.5 Kết nối Firebase vào dự án Lotus – Quản lý kho**

## 3.5.1 Firebase là gì?

Firebase là một nền tảng phát triển ứng dụng web và mobile, cung cấp cho các nhà phát triển một bộ công cụ và dịch vụ giúp họ phát triển các ứng chất lượng cao, tăng trưởng lượng người dùng và kiếm được nhiều lợi nhuận hơn.

## 3.5.2 Các tính năng chính của Firebase

**a. Realtime database**

Firebase Realtime Database là một cơ sở dữ liệu NoSQL trên đám mây cho phép bạn lưu trữ và đồng bộ hóa dữ liệu giữa các user trong thời gian thực.

**b. Authentication**

Firebase Authentication cung cấp dịch vụ backend, SDKs sẵn sàng sử dụng, và các thư viện UI được làm sẵn để giúp ứng dụng của bạn xác thực người dùng.

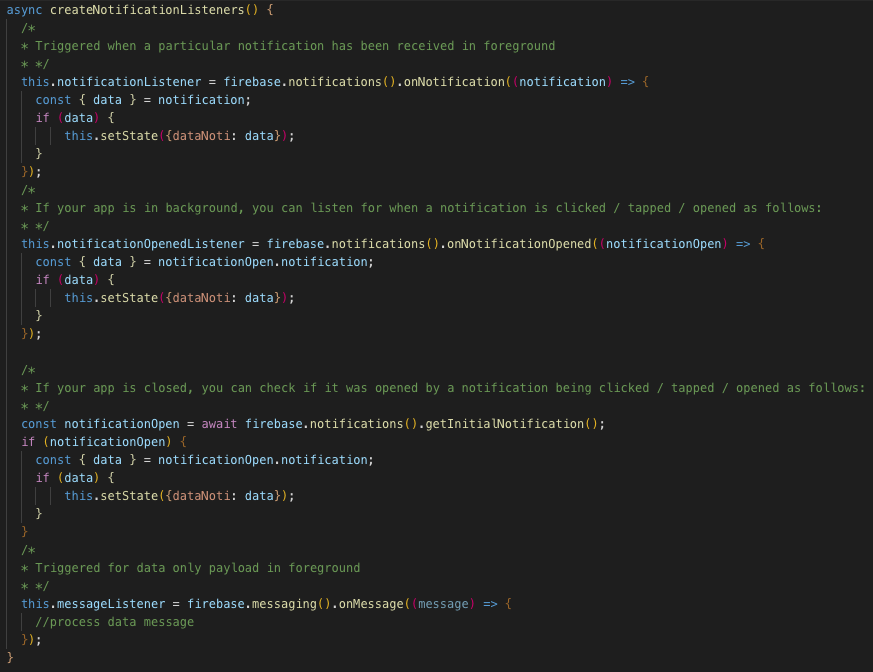
Bạn có thể xác thực người dùng qua các phương thức như: Email, Phone number, Facebook, Google, Twitter, ...

**c. Firebase Cloud Messaging (FCM)**

Firebase Cloud Messaging cung cấp một kết nối hiệu quả và đáng tin cậy giữa server và thiết bị của bạn, cho phép bạn gửi và nhận tin nhắn hoặc thông báo trên iOS, Android và Web mà không mất thêm chi phí.

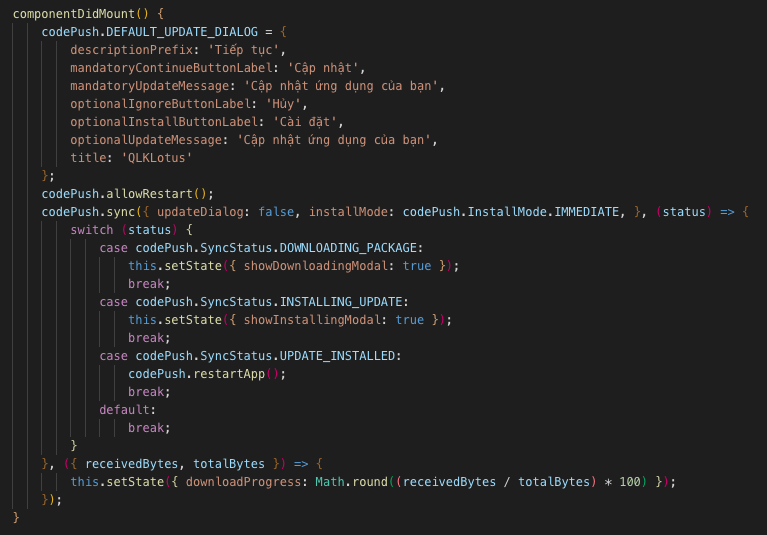
Bạn có thể gửi tin nhắn thông báo (giới hạn 2KB) hoặc tin nhắn dữ liệu (giới hạn 4KB).

## 3.5.3. Cài đặt, cấu hình và cách dùng Firebase

Để dùng được Firebase trong dự án này ta cần cài mỗi thư viện là react-native-firebase vào trong dự án để gửi thông báo cho người dùng. ***Cấu hình firebase khi user nhận được thông báo***

# **3.6 Cách đẩy code lên hệ điều hành** **ios, android dự án Lotus – Quản lý kho**

Để đẩy code lên hệ điều hành ios, android trong dự án này ta cần cài thư viện là react-native-code-push và config như dưới đây:



Để đẩy code lên các hệ điều hành cần phải đúng branch dự án yêu cầu và phải kiểm tra máy áo chạy ổn định trước khi push code lên thiết bị thật.

***code-push release-react QLKLotus-iOS ios -m*** cho hệ điều hành IOS

***code-push release-react QLKLotus-Android android -m*** cho hệ điều hành Android

# **KẾT LUẬN**

Thông qua thời gian thực tập, được tiếp xúc với những thiết bị thực thế và kỹ năng

làm việc chuyên nghiệp, em đã tiếp thu được nhiều kiến thức mới để sẵn sàng cho

những bước tiến mới của bản thân.

Kiến thức lĩnh hội được:

- Cách làm việc nhóm

- Cách phát triển mobile app bằng React Native